Commodore
Sachbuch

Uwe Gerlach
Christian Hochberger

AMIGA-HARDWARE-TUNING

Für Amiga 500, 1000 und 2000
Mit vielen Platinenlayouts und ausführlichen Selbstbauanleitungen: EPROM-Programmiergerät ★ Genlock-Interface ★ Midi-Interface ★ Modem ★ Akkugepufferte RAM/ROM-Disk ★ Digitizer

Auf Diskette enthalten:
Treibersoftware zu allen Hardware-Zusätzen.

Amiga-Hardware-Tuning

Uwe Gerlach
Christian Hochberger

Commodore Sachbuch

AMIGA-HARDWARE-TUNING

Für Amiga 500, 1000 und 2000
Mit vielen Platinenlayouts und ausführlichen
Selbstbauanleitungen ★ EPROM-Programmmiergerät
Genlock-Interface ★ Midi-Interface
Modem ★ Akkugepufferte RAM/ROM-Disk ★ Digitizer

Markt&Technik Verlag AG

CIP-Titelaufnahme der Deutschen Bibliothek

Gerlach, Uwe:
Amiga-Hardware-Tuning : für Amiga 500, 1000 und 2000, mit vielen Platinenlayouts
und ausführlichen Selbstbauanleitungen: EPROM-Programmiergerät,
Genlock-Interface, Midi-Interface, Modem,
akkugepufferte RAM/ROM-Disk, Digitizer /
Uwe Gerlach ; Christian Hochberger. –
Haar bei München : Markt-u.-Technik-Verl., 1989.
(Commodore-Sachbuch)
ISBN 3-89090-586-2
NE: Hochberger, Christian:

Die Platinen wurden vom Verlag nachgebaut und mit der veröffentlichten Software intensiv getestet. Soft- und Hardwarefehler konnten nicht festgestellt werden. Trotzdem haftet der Markt&Technik Verlag nicht für Schäden jeglicher Art, die auf den Einbau zurückzuführen sind. Außerdem bedenken Sie, daß durch das Öffnen des Gerätes die Garantie erlischt. Vorsicht, der Computer wird mit 220 Volt betrieben. Bei geöffnetem Gerät stets die Stromzuführung unterbrechen.

15 14 13 12 11 10 9 8 7 6 5 4 3 2
93 92 91 90

ISBN 3-89090-586-2

Einbandgestaltung: Grafikdesign Heinz Rauner
Druck: Schoder, Gersthofen
Printed in Germany

Inhaltsverzeichnis

Vorwort

Alles begann im Jahre 1982. Bald nachdem in Florida eine Firma namens »Amiga« mit dem Ziel gegründet wurde, einen Heimcomputer für Flugsimulatonen zu entwickeln, der alles bisher dagewesene in den Schatten stellen sollte, ging in Los Gatos eine Entwickler-Crew unter Leitung des ehemaligen Atari Chip-Designers Jay Miner mit vielen neuen Ideen ans Werk.

Zwei Jahre später wurde auf der Consumer Electronics Show eine erste Version des Gerätes vorgestellt, doch die Arbeiten hatten mehr Geld verschlungen, als geplant.

Das Startkapital der kleinen Firma war aufgefressen, obwohl noch kein einziger der geplanten Chips endgültig fertiggestellt war. Man mußte an ein kapitalkräftiges Unternehmen verkaufen.

Bei seiner Vorstellung hatte der Amiga-Computer Eindruck gemacht. Ein Vertrag mit der Firma Atari, einem Ableger des Multi-Media-Konzerns Warner Brothers, war fast unter Dach und Fach, als der Branchenriese Commodore plötzlich ein überraschend hohes Angebot machte, und den Zuschlag erhielt. Das Konzept des Computers wurde nun gründlich überarbeitet. Ein reines Simulationsgerät war kein Thema mehr.

Der Amiga bekam neben allen Einrichtungen, die ein »normaler« Computer so braucht, als vielleicht herausragendste Komponente noch ein solides Multitasking-Betriebssystem.

Am 23. Juli 1985 konnte der aus dieser Verbindung hervorgegangene Supercomputer in New York endlich der Öffentlichkeit präsentiert werden.

Die Entwicklung hatte lange gedauert, doch der Amiga 1000 hatte wirklich überragende Eigenschaften. Allerdings war er zu teuer und konnte sich gegen die inzwischen unter Leitung des ehemaligen Commodore-Chefs Jack Tramiel im Eilverfahren aufgebaute ST-Serie des Konkurrenten Atari nur schwer behaupten.

Mittlerweile hat Commodore Konsequenzen gezogen. Der »alte« Amiga 1000 wurde abgelöst durch zwei »neue« Modelle, nämlich durch den Amiga 500, einer auf das nötigste abgespeckten Version für den Heimcomputermarkt, und durch den Amiga 2000, der deutlich auf professionelle Anwender zielt. Der 500 wurde in

West Chester, Großbritannien entwickelt, der 2000 in Braunschweig, Bundesrepublik Deutschland.

Die Änderungen beschränkten sich aber meist auf Äußerlichkeiten, so daß die Software zum weitaus größten Teil auf allen drei Rechnern läuft. Im Amiga 500 versucht man Kosten einzusparen, indem viele Standardbauelemente noch in eine höher integrierte Version des Spezialchips AGNUS (»FAT AGNUS«) gepackt wurden. Als wichtigste Sparmaßnahme fehlt beiden neuen Computern gegenüber dem Amiga 1000 aber das Kickstart-RAM. Stattdessen wurde ein entsprechendes ROM eingesetzt. Einerseits hat das ein schnelleres und einfacheres Hochfahren des Systems zur Folge, andererseits entfällt dadurch die Möglichkeit, ohne Platzverlust unterschiedliche Betriebssysteme nachladen zu können. Man bleibt auf die eingebaute Kickstart-Version festgelegt.

Durch seine Entwicklungsgeschichte geriet der Amiga zu einem Computer der Superlative, nicht nur, was seine Fähigkeiten in den Bereichen Grafik, Animation und Sound angeht. Bild 1.0 zeigt schematisch den inneren Aufbau eines Amiga. Um den Prozessor 68000 herum gruppieren sich neben dem Speicher und seiner Ansteuerlogik vor allem fünf Baugruppen, die jeweils in einem IC untergebracht sind, nämlich zwei I/O-Bausteine und die drei Amiga-Spezialchips PAULA, DENISE und AGNUS. Besonders sie verleihen dem Computer durch ihre Komplexität eine solche Vielfalt an Möglichkeiten, daß wohl kaum jemals alle voll ausgeschöpft werden können.

Dieses Buch soll wichtige Hardwaredetails der Amiga-Serie erläutern und ihre Verwendung anhand von Beispielen verdeutlichen. Dabei werden viele nachbausichere Schaltungen vorgestellt. Im Anhang sind die Layouts enthalten und eine Diskette mit der jeweiligen Betriebssoftware liegt bei.

Beim ersten Kapitel handelt es sich um eine ausführliche Funktions- und Programmierbeschreibung der in den Amigas eingebauten Schnittstellenbausteine 8520. Zu wissen, welche Funktionen diese Bausteine erfüllen, wie sie im Amiga benutzt werden, und welche Teile noch frei für eigene Verwendung sind, ist sowohl für Hardware-Entwickler, wie auch für Programmierer sehr wichtig.

Das zweite Kapitel enthält Beispielschaltungen für externe Erweiterungen. Hier sehen Sie, wie der Amiga seine Umwelt beeinflussen kann. Das reicht vom Ein- und Ausschalten von Lämpchen bis zum Reagieren auf Schall. Außerdem werden nützliche Zusätze zum Nachbau vorgestellt, wie ein komfortabler EPROMer.

Der dritte Teil des Buches behandelt die drei Amiga-Spezialbausteine und ihre Programmierung anhand von mehreren Beispielen, wie Maus-, Joystick- und Paddle-Abfrage, Handhabung der seriellen Schnittstelle, des Interruptkontrollers und der Diskettensteuerung.

Ein weiteres Kapitel dreht sich um die Video-Hardware. Nach einer kurzen Erläuterung der Video-Grundlagen wird auf spezielle Features beim Amiga hingewiesen und auch ein Selbstbau-Genlock-Interface vorgestellt.

Zum Schluß wird die Erweiterungsmöglichkeit über die Expansionbus-Steckverbinder bzw. die Slots des Amiga 2000 erläutert. Dabei kommt der vorgesehenen Autokonfigurierbarkeit aller Einheiten besonderer Wert zu. Dem Buch liegt eine doppelseitige, durchkontaktierte Leerplatine bei, die vielseitig nutzbar ist, beispielsweise als autokonfigurierende ROM- oder akkugepufferte CMOS-RAM-Disk, von der aus der Rechner nach dem Einschalten ohne Einsatz einer Diskette sehr schnell gebootet wird, als normale Speichererweiterung und als Adapter zum Anschluß einer Harddisk bzw. von PC-Zusatzkarten ohne PC- oder AT-Brückenkarte!

Das Buch enthält viele Tabellen und Schaltpläne. Dabei bezieht es sich immer auf alle drei verfügbaren Ausführungen des Amiga. Im Zuge der Umstrukturierung zum Amiga 500 bzw. 2000 wurden auch die Belegungen der Steckverbinder gegenüber dem Amiga 1000 leicht geändert und dadurch bestehenden Standards angepaßt. Bei Abweichungen wird auf die entsprechenden Punkte hingewiesen. Es werden sogar Adapter vorgestellt, die – soweit möglich – jeden bisher verfügbaren Hardwarezusatz an jedem der Rechner einsetzbar machen.

An dieser Stelle möchten wir der Firma Alphatron danken, die uns ihr Layoutprogramm Newio zur Verfügung gestellt hat. Ebenso gilt unser Dank Bernhard Möllemann, unter dessen Mitarbeit das Memory-Device entstanden ist. Beim Nachbau und Betrieb der Zusatzschaltungen wünschen wir Ihnen nun viel Freude.

Uwe Gerlach
Christian Hochberger

1

Die Amiga-Portbausteine

Jeder Computer benötigt programmierbare Steuerleitungen, einerseits für seine Schnittstellen, die in letzter Zeit immer größere Bedeutung erlangt haben, andererseits aber auch für innere Schaltvorgänge oder Zustandsabfragen. Dieses Kapitel erläutert zunächst die Eigenschaften der im Amiga eingesetzten Portbausteine und zeigt ihre Verwendung im System sowie Möglichkeiten für externe Benutzung auf.

1.1 Doppelt gemoppelt

Commodore setzt in den Amiga-Computern jeweils zwei I/O-Bausteine vom Typ 8520 ein. Diese leistungsfähigen ICs sind nicht völlig neu, sondern lediglich eine Weiterentwicklung der CIA (Complex Interface Adaptor)-Bausteine 6526, die sich bereits im C64 millionenfach bewährt haben. Bild 1.2 zeigt das Pinout des 8520. Es ist mit dem des 6526 identisch. Allerdings wurden intern bei der Echtzeituhr Änderungen vorgenommen. Sollte in Ihrem System ein 8520 defekt sein, läßt er sich zur Not durch einen 6526 ersetzen. Versuche haben dies bestätigt.

Jeder der beiden 8520 im Amiga besitzt – wie bei vergleichbaren ICs allgemein üblich – zwei acht Bit breite Ports zur Ein- und Ausgabe mit Handshake-Möglichkeit.

Ebenfalls gängig sind Timer, von denen der 8520 gleich zwei Stück mit je 16 Bit Breite besitzt, die auch noch – einzeln oder gemeinsam – äußerst universell programmierbar sind. Voraus hat er vergleichbaren Bausteinen die Verfügbarkeit eines seriellen Ports zur wahlweisen Ein- oder Ausgabe von Daten, und quasi als Zugabe besitzt er einen 24-Bit-Zähler mit Alarmstand-Erkennung.

Zur Unterscheidung werden diese vielseitigen Bausteine im Amiga 8520-A und 8520-B genannt. Jedem sind dabei spezielle Funktionen zugeteilt. So ist Baustein A vor allem für die Kontrollsignale der seriellen Schnittstelle und für die Floppy-Steuerung zuständig. Baustein B dagegen bedient neben diversen Funktionen für Game-Ports, Speicherverwaltung, Anzeige und Floppy hauptsächlich die parallele Schnittstelle. Tabelle 1.1 zeigt die genauen Aufgaben jedes Bausteins.

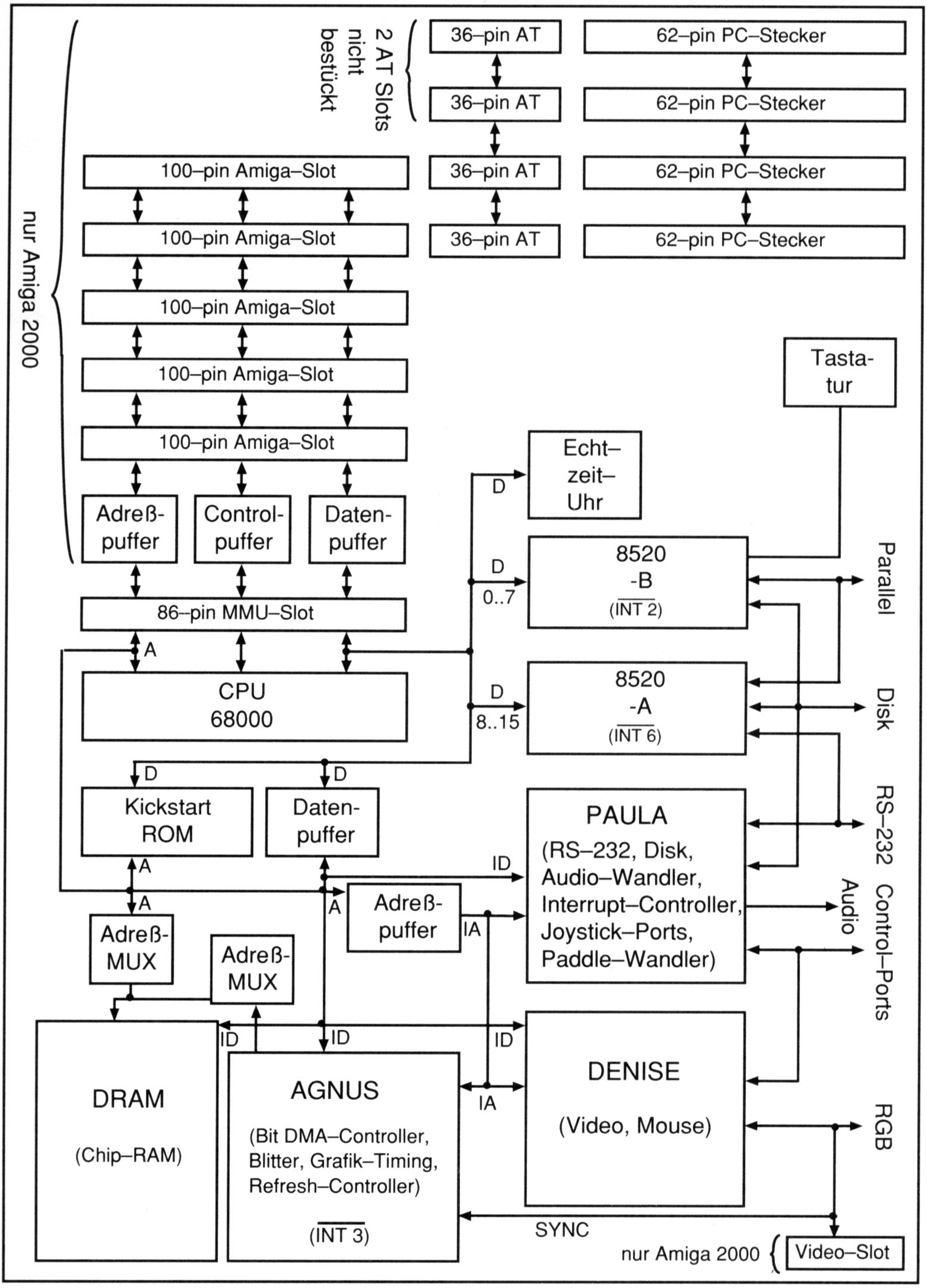


Bild 1.1: Blockschaltbild eines Amiga

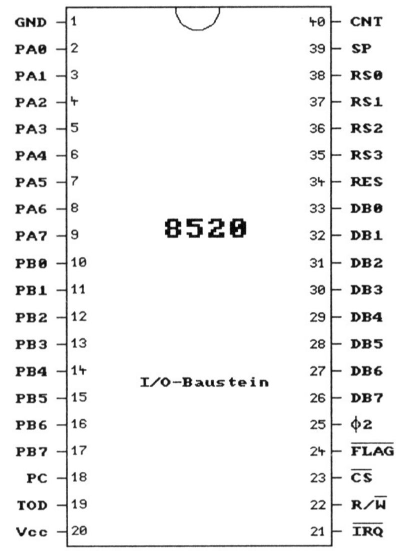


Bild 1.2: Pinbelegung der I/O-Bausteine 8520

1.2 Unter der Haube

Beginnen wir nun, den Schnittstellenbaustein 8520 im einzelnen kennenzulernen.

Seine diversen Funktionen werden über 16 Adreßregister gesteuert, die in Tabelle 1.2 zusammengestellt sind. Die bausteininternen Registeradressen wurden im Amiga nicht aufeinanderfolgenden Memory-Adressen zugeordnet, sondern jedes Register belegt gleich 128 Worte. Dabei ist der 8520-A auf ungeraden, der 8520-B auf geraden Adressen ansprechbar. Die jeweiligen Basisadressen sind in Tabelle 1.2 enthalten.

Prinzipiell lassen sich die Register genau so beschreiben und auslesen, wie ganz normale Speicherzellen auch, also von Basic aus etwa mit PEEK (Adresse) zum Lesen und POKE (Adresse), Wert zum Schreiben. Bei Steuerregistern hat jedoch jedes Bit seine ganz spezielle Bedeutung und es können Unterschiede in der Funktion zwischen Lesen und Schreiben auftreten. Aus diesem Grund sollte man in Maschinenprogrammen nicht die 68000-Befehle BCLR und BSET zum Löschen bzw. Setzen eines Bits verwenden. Der Prozessor müßte für diese Operation zunächst das entsprechende Byte lesen und würde wahrscheinlich von einem falschen Inhalt ausgehen.

1.2.1 Die I/O-Ports

Port heißt eigentlich Tor oder Hafen und assoziiert in übertragenem Sinne Verbindung oder Anschluß.

Für den Prozessor im Amiga stellen die programmierbaren Leitungen ein Tor zur Außenwelt dar.

8520-A:

PA0:	BUSY	par. Schnittst.	Ausgang verbunden mit SP
PA1:	POUT	par. Schnittst.	Eingang verbunden mit CNT
PA2:	SEL	par. Schnittst.	Eingang
PA3:	$\overline{\text{DSR}}$	ser. Schnittst.	Eingang über Treiber (MC1489A)
PA4:	$\overline{\text{CTS}}$	ser. Schnittst.	Eingang über Treiber (MC1489A)
PA5:	$\overline{\text{CD}}$	ser. Schnittst.	Eingang über Treiber (MC1489A)
PA6:	$\overline{\text{RTS}}$	ser. Schnittst.	Ausgang über Treiber (MC1488)
PA7:	$\overline{\text{DTR}}$	ser. Schnittst.	Ausgang über Treiber (MC1488)
PB0:	$\overline{\text{STEP}}$	Floppy-Signal	Step
PB1:	$\overline{\text{DIR}}$	Floppy-Signal	Direction
PB2:	$\overline{\text{SIDE}}$	Floppy-Signal	Side Select
PB3:	$\overline{\text{SEL0}}$	Floppy-Signal	Select Internal Drive
PB4:	$\overline{\text{SEL1}}$	Floppy-Signal	Select External 1st Drive
PB5:	$\overline{\text{SEL2}}$	Floppy-Signal	Select External 2nd Drive
PB6:	$\overline{\text{SEL3}}$	Floppy-Signal	Select External 3rd Drive
PB7:	$\overline{\text{MTR}}$	Floppy-Signal	Motor
SP:	BUSY	par. Schnittst.	Ausgang verbunden mit PA0
CNT:	POUT	par. Schnittst.	Eingang verbunden mit PA1
PC:		(nicht verbunden)	
$\overline{\text{FLAG}}$:	$\overline{\text{INDEX}}$	Floppy-Signal	Disk Index
TOD:	$\overline{\text{BHS}}$	Eingang	gepuffertes HSync zur Sprite-Darstellung
Timer A:		normalerweise unbenutzt	
Timer B:		Video Beam Follower (zur Synchronisation des Blitters auf den Video-Strahl. Frei, falls keine Blitter-Synchronisation aktiv.)	

8520-B:

PA0:	OVL	intern	Memory Overlay Bit
PA1:	$\overline{\text{LED}}$	Betriebsanzeige	(leuchtet bei 0)
PA2:	$\overline{\text{CHNG}}$	Floppy-Signal	Disk Change
PA3:	$\overline{\text{WPRO}}$	Floppy-Signal	Write Protect
PA4:	$\overline{\text{TK0}}$	Floppy-Signal	Disk Track 00
PA5:	$\overline{\text{RDY}}$	Floppy-Signal	Disk Ready
PA6:	$\overline{\text{GAME PORT 0}}$, Pin 6		(Feuerknopf)
PA7:	$\overline{\text{GAME PORT 1}}$, Pin 6		(Feuerknopf)

PB0:	P0	par. Schnittst.	Data 0
PB1:	P1	par. Schnittst.	Data 1
PB2:	P2	par. Schnittst.	Data 2
PB3:	P3	par. Schnittst.	Data 3
PB4:	P4	par. Schnittst.	Data 4
PB5:	P5	par. Schnittst.	Data 5
PB6:	P6	par. Schnittst.	Data 6
PB7:	P7	par. Schnittst.	Data 7
SP:	KDAT	Keyboard	Data
CNT:	KCLK	Keyboard	Clock
$\overline{\text{PC}}$:	$\overline{\text{DRDY}}$	par. Schnittst.	Eingang $\overline{\text{Strobe}}$
$\overline{\text{FLAG}}$:	$\overline{\text{ACK}}$	par. Schnittst.	Ausgang Acknowledge
TOD:	TICK	Eingang	50/60Hz-Signal für Echtzeituhr
Timer A:		Tastatur (Immer belegt, wenn SYSTEM EXEC aktiv)	
Timer B:		Virtual Timer Device (zur Umschaltung der Tasks und Interrupterzeugung. Immer belegt, wenn SYSTEM EXEC aktiv.)	

Tabelle 1.1: Verwendung der beiden I/O-Bausteine 8520 im Amiga

Der 8520 besitzt zwei 8-Bit-Ports, die jeweils von zwei Registern kontrolliert werden, nämlich Port A (PA0 – PA7) von Register 0 und 2, Port B (PB0 – PB7) dagegen von Register 1 und 3. Jede einzelne Leitung kann entweder als Eingang oder als Ausgang programmiert werden. Es ist also möglich, an einen Pin programmgesteuert eine logische Spannung anzulegen, oder es können die außen anliegenden Spannungen als logische Werte (HIGH oder LOW) in den Rechner eingelesen werden. Spannungen nahe 0 Volt ergeben logisch 0 oder LOW, Spannungen nahe 5 Volt logisch 1 bzw. HIGH.

1.2.1.1 Das Datenrichtungsregister

Die Festlegung, welche Leitung Eingang und welche Ausgang sein soll, geschieht über das Datenrichtungsregister (Data Direction Register DDR...).

Für Port A ist die Bausteinadresse 2 (DDRA) zuständig, Port B reagiert auf Adresse 3 (DDRB). Darin steht jedes Bit für eine bestimmte Leitung. Bit 0 steuert PB0, Bit 1 PB1 und so weiter. Ist ein Bit gesetzt (1), so wird die entsprechende Leitung zum Ausgang. Ein gelöschtes Bit programmiert sie auf Eingang.

1.2.1.2 Das Datenregister

Ganz analog sind die Bits des Datenregisters (Port Register PR...) den einzelnen Leitungen zugeordnet. Das Datenregister für Port A (PRA) hat die Bausteinadresse 0, das für Port B (PRB) die Adresse 1. Ist ein Kanal über das Datenrichtungsregister auf Ausgabe programmiert, erzeugt ein gesetztes Bit auf der entsprechenden Ausgabeleitung eine Spannung mit HIGH-Pegel (nahe +5 Volt), ein gelöschtes Bit jedoch LOW-Pegel (nahe 0 Volt).

Jede Portleitung des 8520 kann zwei TTL-Lasten treiben.

1.2.2 Datenaustausch mit Händeschütteln

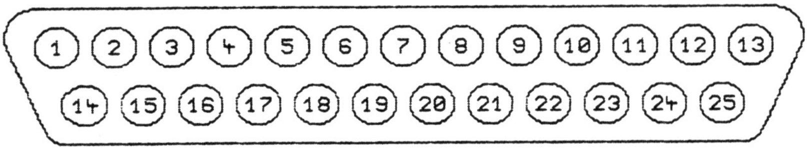
Zwei miteinander verbundene Geräte, beispielsweise ein Computer und ein Drucker, werden im Normalfall nicht exakt die gleiche Arbeitgeschwindigkeit haben.

Es genügt also nicht, eine Folge von Bitkombinationen nacheinander auf die Ver-

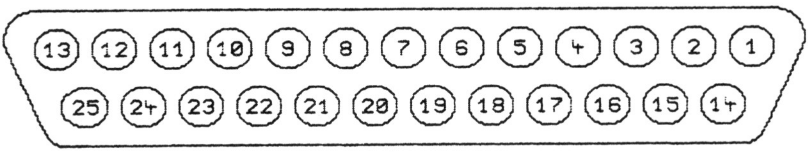
Int.Adr.	Mem.-Adresse	Name	Funktion bei Schreiben	Funktion bei Lesen	Bemerkungen
0	A: $BFE001 B: $BFD000	PRA	(Port Register A) Programmieren der Ausgangspegel LOW bei jew. Bit=0, HIGH bei Bit=1	Eingänge: Pegel der Anschlüsse	Parallelport Maximal 2 TTL-Lasten zulässig
1	A: $BFE101 B: $BFD100	PRB	(Port Register B) Programmieren der Ausgangspegel LOW bei jew. Bit=0, HIGH bei Bit=1	Eingänge: Pegel der Anschlüsse	
2	A: $BFE201 B: $BFD200	DDRA	(Data Direction Register B = Datenrichtungsregister B) Lesen der E/A-Programmierung LOW bei jew. Bit=0, HIGH bei Bit=1	Programmieren als Ein/Ausgänge	
3	A: $BFE301 B: $BFD300	DDRB	(Data Direction Register B = Datenrichtungsregister B) Programmieren als Ein/Ausgänge LOW bei jew. Bit=0, HIGH bei Bit=1	Lesen der E/A-Programmierung	
4	A: $BFE401 B: $BFD400	TALO	(Timer A LOW-Byte) Startwert für Abwärtszählen	Augenblicklicher Zählerstand	16-Bit-Timer
5	A: $BFE501 B: $BFD500	TAHI	(Timer A HIGH-Byte) Startwert für Abwärtszählen	Augenblicklicher Zählerstand	
6	A: $BFE601 B: $BFD600	TBLO	(Timer B LOW-Byte) Startwert für Abwärtszählen	Augenblicklicher Zählerstand	16-Bit-Timer
7	A: $BFE700 B: $BFD701	TBHI	(Timer B HIGH-Byte) Startwert für Abwärtszählen	Augenblicklicher Zählerstand	
8	A: $BFE801 B: $BFD800	LSB Event	(Ereigniszähler, niederwertiger Teil) Stellen der niederwertigen 8 Bit Reg. F, Bit 7: 0:Zähler/1:Alarm	Lesen der niederwertigen 8 Bit immer aktuellen Zählerinhalt	24-Bit-Zähler mit Alarmregister verwendbar als Echtzeit-Uhr
9	A: $BFE901 B: $BFD900	Event 8-15	(Ereigniszähler Bits 8 – 15) Stellen der mittleren 8 Bit Reg. F, Bit 7: 0:Zähler/1:Alarm	Lesen der mittleren 8 Bit immer aktuellen Zählerinhalt	
A	A: $BFEA01 B: $BFDA00	MSB Event	(Ereigniszähler, höherwertiger Teil) Stellen der höchstwertigen 8 Bit Reg. F, Bit 7: 0:Zähler/1:Alarm	Lesen der höchstwerigen 8 Bit immer aktuellen Zählerinhalt	
B	A: $BFEB01 B: $BFDB00		(Unbenutzt)		

Int.Adr.	Mem.-Adresse	Name	Funktion bei Schreiben	Funktion bei Lesen	Bemerkungen
C	A: $BFEC01 B: $BFDC00	SDR	(Serial Data Register = Serielles Datenregister) Schieberegister für Pin SP Ausg. Höchstwertiges Bit erscheint zuerst	Schieberegister für Pin SP Eing.	Serieller Port
D	A: $BFED01 B: $BFDD00	ICR	(Interrupt Control Register = Interrupt-Kontrollregister) Interruptmaskierung für Funktion: Bit 0: Unterlauf Timer A aufgetreten Bit 1: Unterlauf Timer B aufgetreten Bit 2: Event-Zähler hat Alarmstellung erreicht Bit 3: Shiftregister voll (Eingang) bzw. leer (Ausgang) Bit 4: negative Flanke am Pin-FLAG aufgetreten Bit 5 - 6: immer 0 Bit 7 = 1: 1-Bits machen scharf = 0: 1-Bits entschärfen Nur scharfe Bits erzeugen Interr.	jew. Bit=1 falls Bedingung wahr 1=mindestens eine Bedingung bei scharfem Maskenbit aufgetreten Bits werden bei Lesen gelöscht!	Steuerung der Interrupt-Erzeugung
E	A: $BFEE01 B: $BFDE00	CRA	(Control Register A = Kontrollregister A) Bit 0: 0: Timer A Stop, 1: Timer A Start Bit 1: 1: Signalisierung von Unterlauf Timer A an PB6 wie folgt: Bit 2: 0: jeder Unterlauf von A kippt PB6 in die andere Lage 1: jeder Unterlauf erzeugt an PB6 einen HIGH-Impuls von 1μs Bit 3: 0: Timer A zählt fortlaufend vom Ausgangswert (Reg.4/5) abw. 1: Timer A zählt nur einmal vom Ausgangswert (Reg.4/5) auf 0 Bit 4: 0: keine Funktion 1: Register 4/5 wird als Startwert für Timer A übernommen Bit 5: Timertrigger: 0: Timer A zählt Systemtaktpulse 1: Timer A zählt steigende Flanken an Pin CNT Bit 6: 0: SP ist Schieberegistereingang, 1: SP ist Ausgang Bit 7: (unbenutzt)		Steuerung von Timer A
F	A: $BFEF01 B: $BFDF00	CRB	(Control Register B = Kontrollregister B) Bit 0: 0: Timer B Stop, 1: Timer B Start Bit 1: 1: Signalisierung von Unterlauf Timer B an PB7 wie folgt: Bit 2: 0: jeder Unterlauf von B kippt PB7 in die andere Lage 1: jeder Unterlauf erzeugt an PB7 einen HIGH-Impuls von 1μs Bit 3: 0: Timer B zählt fortlaufend vom Ausgangswert (Reg.6/7) abw. 1: Timer B zählt nur einmal vom Ausgangswert (Reg.6/7) auf 0 Bit 4: 0: keine Funktion 1: Register 6/7 wird als Startwert für Timer B übernommen Bit 5, 6: Timertrigger: 00: Timer B zählt Systemtaktpulse 01: Timer B zählt steigende CNT-Flanken 10: Timer B zählt Unterläufe von Timer A 11: Timer B zählt nur dann Unterläufe von A, wenn Pin CNT HIGH ist Bit 7: Schreiben in Reg. 8-A stellt bei 0: Zähler, bei 1: Alarmreg.		Steuerung von Timer B

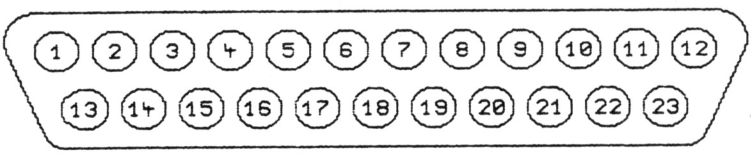
Tabelle 1.2: Registerübersicht für den I/O-Baustein 8520



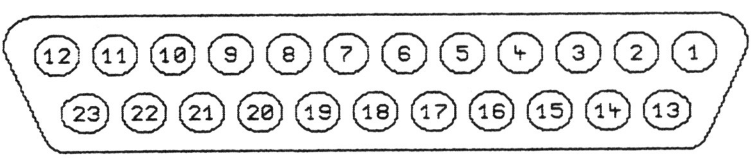
Sub-D, 25-pol, männlich, Draufsicht Steckseite



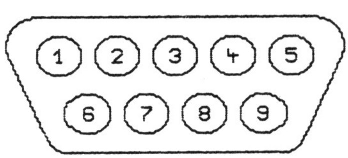
Sub-D, 25-pol, weiblich, Draufsicht Steckseite



Sub-D, 23-pol, männlich, Draufsicht Steckseite



Sub-D, 23-pol, weiblich, Draufsicht Steckerseite



Sub-D, 9-pol, männlich,
Draufsicht Steckerseite

bindungsleitungen zu legen, und sich darauf zu verlassen, daß sie am anderen Ende schon richtig erkannt werden. Vielmehr müssen die beiden verbundenen Geräte Hand in Hand arbeiten. Dazu stellt man jeweils eine Meldeleitung bereit. Die Übertragung zwischen zwei Geräten läuft dann in der Regel folgendermaßen ab:

1. Der Sender legt die Kombination für das zu übermittelnde Zeichen an die Datenleitungen und

2. gibt auf seiner Meldeleitung einen kurzen Impuls aus, um zu signalisieren, daß neue Daten anliegen.

3. Der Empfänger holt sich daraufhin die Daten ab, verarbeitet sie und

4. teilt seine Empfangsbereitschaft durch einen kurzen Impuls auf einer anderen Meldeleitung dem Partner mit.

Danach fängt der nächste Zyklus wieder bei Punkt 1 an.

Dieses Verfahren wird Handshake-Übertragung genannt. (Handshake = Händeschütteln) Die beiden Geräte können intern mit völlig verschiedenen Geschwindigkeiten arbeiten, ohne daß Probleme auftreten, da die Abtastzeitpunkte durch

Schauen Sie sich bitte in Tabelle 1.2 das Interruptkontrollregister (Bausteinadresse 13) genauer an. Bit 4 wird automatisch gesetzt, sobald am Anschluß $\overline{\text{FLAG}}$ des entsprechenden Bausteins ein HIGH-LOW-Übergang aufgetreten ist. Damit steht ein leistungsfähiger Melde-Eingang zur Verfügung. Ein dort eintreffender Impuls – so kurz er auch sein mag – setzt in jedem Fall über ein internes Flipflop das erwähnte Bit 4. Der Computer ist dadurch nicht darauf angewiesen, ständig in möglichst kurzen Zeitabständen einen Eingang abzufragen, sondern kann etwas ganz anderes tun. Wie der Registername schon andeutet, ist sogar eine Interrupterzeugung durch das gesetzte Bit vorgesehen. Damit werden externe Abläufe sehr einfach überwachbar.

Zu beachten ist jedoch, daß jeder Lesezugriff auf das Interruptkontrollregister alle darin befindlichen Informationsbits automatisch löscht. Dieser Effekt ist im allgemeinen sehr nützlich und spart bei richtigem Einsatz Programmieraufwand und Zeit. Zum Löschen des Registers vor der Benutzung ist beispielsweise lediglich eine Leseoperation nötig.

Die Meldeleitung des 8520 ist der Anschluß $\overline{\text{PC}}$. Er ist fest verdrahtet, also

Bit 4:	1=unabhängig davon, ob der Timer gerade läuft oder nicht, wird das Zählregister mit den Werten aus $XX06 und $XX07 geladen 0=keine Funktion
Bit 5, Bit 6:	Triggerquelle für Timer B: 00=Timer B zählt Systemtakte 10=Timer B zählt steigende CNT-Flanken 01=Timer B zählt Unterläufe von Timer A 11=Timer B zählt nur Unterläufe von Timer A, wenn CNT High ist
Bit 7:	Schreibzugriffe auf Register $XX08 bis $XX0B gehen ins 1=Alarmregister 0=Uhrzeitregister

Tabelle 1.3: Die Kontrollregister des 8520

gister 4 und 5 für Timer A, bzw. 6 und 7 für Timer B zuletzt festgelegt wurde. Sofern Bit 3 gelöscht ist, geht der Zählvorgang anschließend normal weiter. War dieses Bit jedoch gesetzt, bleibt der Zähler auf dem programmierten Ausgangswert stehen.

Der Nulldurchgang wird auch als Unterlauf (englisch: Underflow) bezeichnet. Er kann auf verschiedene Weisen signalisiert werden. Zunächst mittels der Portleitungen PB6 bei Timer A, bzw. PB7 bei Timer B. Dazu muß Bit 1 gesetzt sein.

Zu beachten ist, daß bei dieser Betriebsart die Portleitungen PB6 bzw. PB7 immer Ausgänge sind, unabhängig von der Einstellung im Datenrichtungsregister!

Sie werden sich vielleicht schon gefragt haben, woher die Timer ihren Takt bekommen. Mit Bit 5 im Kontrollregister A bzw. Bit 5 und 6 im Kontrollregister B sind für die beiden Zähler verschiedene Quellen programmierbar. Grundsätzlich wird an die Portbausteine im Amiga die durch zehn geteilte Systemtaktfrequenz, also 0,716 MHz, angelegt. Timer A kann nur von dieser Frequenz oder aber von außen her angesteuert werden. Dagegen ist Timer B wesentlich flexibler. Er läßt sich auch mit Unterläufen seines Kollegen Timer A takten, und das sogar wahlweise unter der Bedingung, daß der externe Anschluß CNT gleichzeitig auf HIGH liegt. Mit dieser Funktion ist es möglich, die beiden Zähler zu kaskadieren, also quasi zu einem 32-Bit-Zähler zusammenzukoppeln. Die maximale Stellung ist dann dezimal 4294967294.

Taktet man diesen zusammengeschalteten Zähler mit der internen Frequenz, dann lassen sich Verzögerungszeiten von über 1,6 Stunden erzielen.

Nützlich für Programmbeeinflussungen ist auch die Signalisierungsmöglichkeit eines Unterlaufs im Interrupt-Kontrollregister D, dessen Funktion im folgenden näher erläutert werden soll.

Register D:	(Interrupt-Kontrollregister)
Bit 0:	Unterlauf Timer A
Bit 1:	Unterlauf Timer B
Bit 2:	Uhrzeit = Alarmzeit
Bit 3:	SDR voll/leer (je nach Datenrichtung)
Bit 4:	negative Flanke am Pin $\overline{\text{FLAG}}$ aufgetreten
Bit 5:	(Immer 0)
Bit 6:	(Immer 0)
Bit 7:	Übereinstimmung mindestens eines Bits mit der Interruptmaske

Tabelle 1.4: Das Interrupt-Kontrollregister beim Lesen

1.2.4 Bitte ruhig stören!

Die Bits 0 bis 4 im Interrupt-Kontrollregister D zeigen den Zustand der verschiedenen Funktionseinheiten des CIA-Bausteins an. Sie werden gesetzt, wenn das zugehörige Ereignis stattgefunden hat. Durch Lesen dieses Registers werden jedoch alle Bits gelöscht!

Die Funktion der einzelnen Bits zeigt Tabelle 1.4.

Die Interruptmaske ist eine Bitkombination, die durch Schreiben in dasselbe Register D festgelegt wird. 0-Bits lassen die entsprechenden Stellen des Interrupt-Kontrollregisters unbeeinflußt. Für die 1-Bits legt Bit 7 fest, ob gesetzt oder gelöscht werden soll. Ist Bit 7 gelöscht, löschen auch alle 1-Bits das entsprechende Maskenbit, ist Bit 7 dagegen gesetzt, dann setzen alle 1-Bits das entsprechende Maskenbit.

Falls ein Registerbit und ein Maskenbit gleichzeitig gesetzt sind, so wird zusätzlich das Registerbit 7 gesetzt und der Open-Kollektor-Bausteinanschluß $\overline{\text{IRQ}}$ nach Masse gezogen. Beim Amiga ist der entsprechende Anschluß des 8520-A mit $\overline{\text{INT6}}$ des Interrupt-Kontrollers in PAULA verbunden, der vom 8520-B dagegen mit $\overline{\text{INT2}}$. PAULA faßt diese Interrupt-Eingänge mit anderen Interrupt-Signalen zusammen, überprüft nochmals ihre Berechtigung und löst gegebenenfalls einen Prozessorinterrupt der entsprechenden Ebene aus.

Diese Möglichkeit nutzt das Multitasking-Betriebssystem des Amiga ausgiebig, beispielsweise indem mittels Timer B im 8520-B ständig Interrupts zum Umschalten der parallel laufenden Tasks erzeugt werden.

Damit wäre auch der Begriff Timer (Zeitgeber) für die Zähler geklärt. Genaugenommen kennzeichnet dieser Ausdruck aber nur eine Einsatzmöglichkeit unter vielen.

Mehr zur Interrupt-Behandlung im Amiga enthält Kapitel 3.

1.2.5 Es wird Zeit

Wie schon gesagt, wurde die Echtzeituhr im 8520 gegenüber dem 6526 geändert.

Anstelle eines BCD-Zählers arbeitet hier nur noch ein 24-Bit-Binärzähler, der über drei Register zu lesen bzw. zu stellen ist. Bausteinadresse 8 verarbeitet die acht niederwertigen Bit, Adresse 9 die acht mittleren und Register A die acht höchstwertigen. Adresse B schließlich bleibt unbenutzt.

Seinen Takt erhält der Zähler über den Anschluß TOD (Time Of Day). Jede dort auftretende positive Flanke erhöht den Zählerinhalt um 1. Der Zähler wurde durch die Änderungen universell verwendbar. Im 8520-B des Amiga bezieht TOD zum Beispiel seine Impulse vom Ausgang TICK des Netzgerätes. Damit wird eine netzfrequente Echtzeituhr erreicht. Bekanntlich ist die Netzfrequenz bei uns über lange Zeit betrachtet sehr stabil und eignet sich damit hervorragend für eine solche Anwendung. Im Amiga B2000 ist der netzsynchrone Takt zusätzlich über einen Jumper auf die Video-Vertikalfrequenz umschaltbar. Sie wird über die Systemtaktfrequenz durch Teilung erzeugt und beträgt ebenfalls 50Hz. Diese Modifikationsmöglichkeit ist für Länder interessant, in denen die Netzfrequenz nicht stabil gehalten wird.

Der Zähler im 8520-A dagegen wird mit den Horizontal-Synchronsignalen des Videoteils getriggert und erzeugt Hilfssignale für die Sprite-Darstellung.

Um die laufende Zählerstellung sauber lesen und stellen zu können, ist eine ganz bestimmte Reihenfolge einzuhalten. Es könnte ja passieren, daß während des Lesens ein Übertrag von einer Stelle zur nächsten auftritt. Ein falsches Ergebnis wäre die Folge. Daher wird der aktuelle Zählerinhalt zwischengespeichert, sobald auf eines der Register zugegriffen wird. Intern läuft der Zähler aber weiter. Der Registerinhalt bleibt so lange gespeichert, bis ein Zugriff auf das LSB-Event-Register (Bausteinadresse 8) erfolgt.

Der 8520 verfügt weiterhin über eine Alarm-Möglichkeit. Falls die Alarm-Vorgabe mit dem aktuellen Zählerinhalt übereinstimmt, wird automatisch Bit 2 im Interrupt-Kontrollregister (Bausteinadresse D) gesetzt. Um die Alarm-Zählerstellung zu programmieren, kommen ebenfalls die genannten Register zum Einsatz. Allerdings ist vorher Bit 7 im Kontrollregister B (Bausteinadresse F) zu setzen. Die Alarmzeit kann nur gestellt werden. Bei einem Lesezugriff erhält man immer den aktuellen Zählerstand.

1.2.6 Bit für Bit im Gänsemarsch

Der serielle Port taucht in Tabelle 1.2 an drei Stellen auf. Zunächst bildet die Bausteinadresse C das serielle Datenregister. Bit 6 im Kontrollregister A (Bausteinadresse E) legt die Richtung der Datenübertragung fest und schließlich gibt Bit 3 des Interrupt-Kontrollregisters (Bausteinadresse D) den Zustand des Datenports wieder. Außerdem spielt noch Timer A eine Rolle.

Der Pin SP stellt den seriellen Portanschluß dar, der Pin CNT führt das zugehörige Taktsignal. Die Programmierung des seriellen Ports auf Ausgang geschieht – wie bei allen anderen Portleitungen auch – durch Setzen des Richtungsbits auf

1. Nach dem Einschalten oder nach einem Reset ist dieses Bit immer gelöscht und der Port somit als Eingang definiert.

Das serielle Datenregister ist ein acht Bit breiter Pufferspeicher. Der eigentliche Datenaustausch geschieht über ein Schieberegister, das vom Programmierer nicht erreichbar ist.

Falls der serielle Port als Eingang programmiert ist, fungiert auch CNT als Takteingang. Bei jeder positiven Flanke an CNT wird der Zustand von SP in das Schieberegister übernommen. Dabei wird der bisherige Inhalt dieses Registers nach links geschoben und das neue Bit rechts angefügt. Ist dies achtmal geschehen, wird der Wert des Schieberegisters in das serielle Datenregister übertragen. Gleichzeitig wird Bit 3 im Interrupt-Kontrollregister gesetzt, um anzuzeigen, daß ein neues Datenwort bereitsteht.

Im Ausgabemodus dient der Anschluß CNT als Taktausgang. Bei jedem Unterlauf von Timer A ändert er seinen Zustand. Mit Timer A wird also die Geschwindigkeit der Datenausgabe bestimmt. Da der kleinstmögliche Inhalt des Timers $0001 ist ($0000 erzeugt keinen Unterlauf!), beträgt die höchste Übertragungsrate ein Viertel des Systemtaktes. Schreibt man einen Wert in das serielle Datenregister, wird er in das interne Schieberegister übernommen, sobald dieses leer ist. Der Inhalt des Schieberegisters wird Bit für Bit auf den Anschluß SP gelegt und dann eine positive Flanke an CNT ausgegeben. Dabei ist auch hier die Schieberichtung wieder links, so daß das höchstwertige Bit zuerst erscheint. Sind alle acht Bits herausgeschoben, so wird – analog zum Empfang von Daten – Bit 3 im Interrupt-Kontrollregister gesetzt. Dies geschieht nicht, wenn bereits vor der Übertragung des achten Bits neue Daten ins serielle Datenregister geschrieben wurden. In diesem Fall würde die Übertragung direkt fortgesetzt. Falls kein neues Byte ins Datenregister geschrieben wird, hält die Datenausgabe an.

SP und CNT besitzen beide als Ausgänge Open-Collector-Stufen. Beim 8520-B sind die Pull-up-Widerstände im Amiga eingebaut, beim 8520-A nicht.

1.2.6.1 »Gerätchenfrage«

Im Amiga werden die seriellen Ports der 8520-Bausteine nicht etwa zur Bedienung der RS-232-Schnittstelle eingesetzt. Dazu fehlen wichtige Merkmale, wie beispielsweise automatisch einfügbare Start- und Stopbits. Zu diesem Zweck enthält der Spezialchip PAULA entsprechende Innereien. Lediglich die Kontrollsignale der RS-232-Schnittstelle bearbeitet der 8520-A über seine I/O-Leitungen PA3 bis PA7. Sein serieller Portanschluß ist, allerdings zusammengeschaltet mit PA0, am Steckverbinder der parallelen Schnittstelle (BUSY) zugänglich. Ebenso der zugehörige Anschluß CNT zusammen mit PA1 (POUT).

Der 8520-B stellt mit seinem seriellen Port die Verbindung zum Tastaturprozessor her. Bild 1.2 zeigt im oberen Teil die Verschaltung. SP und CNT sind jeweils über einen Pull-up-Widerstand zum Keyboard-Connector geführt. Gleichzeitig beeinflußt CNT die Reset-Logik des Amiga.

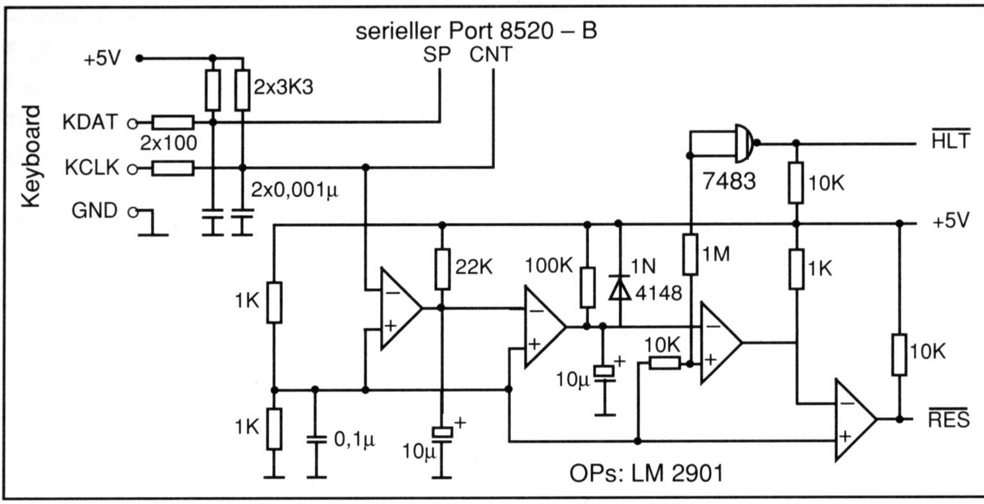


Bild 1.3: Die Reset-Logik mit Beeinflussungsmöglichkeit durch CNT

Die kompliziert aussehende Schaltung mit den vier Operationsverstärkern hat nur die Aufgabe, beim Einschalten der Versorgungsspannung sowie bei längeren LOW-Zeiten von CNT einen Systemreset herbeizuführen. Die Tastatur verfügt dazu über die Möglichkeit, eine solche LOW-Zeit auszulösen.

In der Tastatur arbeitet ein 6500/1-Prozessor mit vier 8-Bit-I/O-Ports, einem 16-Bit-Timer, 2 Kilobyte ROM und 64 Byte RAM. Die Datenübertragung über den seriellen Port erfolgt mit etwa 17 Kilobit pro Sekunde. Das dürfte auch für Schnelltipper mehr als ausreichen.

Werden gleichzeitig die beiden Amiga-Tasten und CTRL gedrückt, zieht der Tastaturprozessor die Taktleitung KCLK für 500 Millisekunden nach Masse. Sind die Tasten danach immer noch gedrückt, verlängert sich die Zeit um weitere 500 Millisekunden, so lange, bis mindestens eine Taste freigegeben wurde.

Dann führt er einen Keyboard-Reset durch.

Im Amiga reichen 500 Millisekunden LOW-Zeit, um einen Systemreset auszulösen. Natürlich läßt sich auch über den Computer ein solcher Hard-Reset herbeiführen, indem von dort aus der Anschluß CNT vom 8520-B entsprechend lang LOW gemacht wird.

Daß der Amiga 1000 nicht ein zweitesmal Kickstart bootet, liegt an einer Schutzkennung, die mit dem ersten Beschreiben des Kickstart-RAM-Bereichs gesetzt wurde. Der Amiga erkennt daran, daß sein Betriebssystem bereits vorhanden ist.

2 Schaltungen am Parallelport

Nachdem Sie nun wissen, wie bestimmte Leitungen am Computer programmgesteuert geschaltet werden können, sollen einige Anwendungsbeispiele folgen. Es handelt sich zunächst um einfache Ein- und Ausgabeschaltungen, die leicht mit Standardbauelementen nachzubauen sind, und auch wie Module in eigene Schaltungen eingesetzt werden können. Danach werden größere Projekte vorgestellt, unter anderem eine I²C-Schnittstelle, ein Digitizer und ein EPROM-Programmierer.

2.1 Der Parallelport

Zum Anschluß von kleinen Zusatzschaltungen wird meist der Parallelport Verwendung finden. Im weiteren Verlauf des Buches werden auch noch andere Steuerleitungen beschrieben. So die Control-Ports in Kapitel 3.3.4.

2.1.1 Anschluß gesucht

Leider wurde die Steckerbelegung beim Amiga 500 bzw. 2000 gegenüber der des Amiga 1000 leicht geändert, um völlig kompatibel zu den entsprechenden Steckverbindungen bei IBM zu werden. Tabelle 2.1 zeigt daher die Steckerbelegung bei den neuen Modellen und Tabelle 2.2 die beim Amiga 1000.

Berücksichtigt man, daß beim Amiga 1000 im Gerät selbst ein männlicher 25-poliger Sub-D-Stecker Verwendung findet, in den übrigen Geräten dagegen

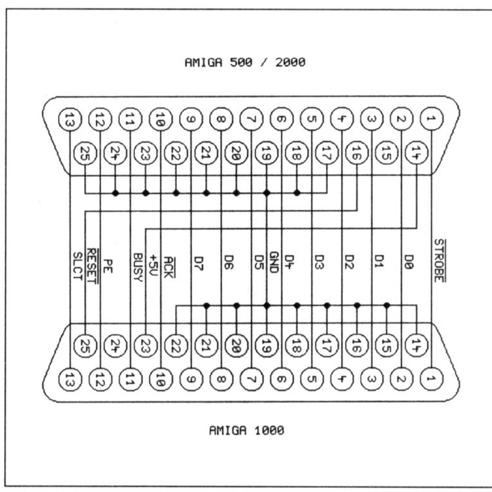
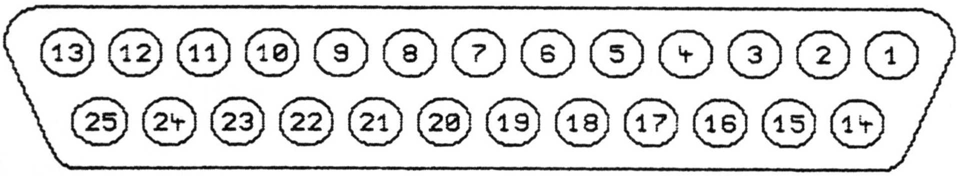
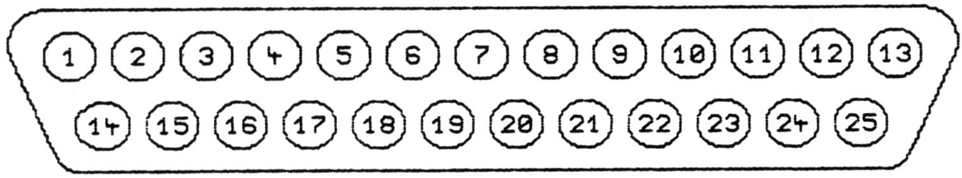


Bild 2.1: Verdrahtung eines Adapters zum Anschluß von für den Parallelport des Amiga 1000 gedachten Zusätzen an den Amiga 500 bzw. 2000 (beide Stecker männlich) oder solchen für den Amiga 500 bzw. 2000 an den Amiga 1000 (beide Stecker weiblich)



Pin	Signal	Funktion	intern	Richtg.
1	$\overline{\text{STROBE}}$	Strobe-Signal	8520-B $\overline{\text{PC}}$	aus
2	D0	Datenbit 0	8520-B PB0	ein/aus
3	D1	Datenbit 1	8520-B PB1	ein/aus
4	D2	Datenbit 2	8520-B PB2	ein/aus
5	D3	Datenbit 3	8520-B PB3	ein/aus
6	D4	Datenbit 4	8520-B PB4	ein/aus
7	D5	Datenbit 5	8520-B PB5	ein/aus
8	D6	Datenbit 6	8520-B PB6	ein/aus
9	D7	Datenbit 7	8520-B PB7	ein/aus
10	$\overline{\text{ACK}}$	Acknowledge	8520-B $\overline{\text{FLAG}}$	ein
11	BUSY	Drucker aktiv	8520-A PA0	ein/aus
			8520-A SP	ein/aus
12	POUT	Papierende	8520-A PA1	ein/aus
			8520-A CNT	ein/aus
13	SEL	Drucker on-line	8520-A PA2	ein/aus
14	+5V	Versorgungsspannung	+5V (47 Ohm)	
15	---	(nicht belegt)		
16	$\overline{\text{RESET}}$	Reset-Leitung	gepuffert	aus
17	GND	Signalmasse	GND	
18	GND	Signalmasse	GND	
19	GND	Signalmasse	GND	
20	GND	Signalmasse	GND	
21	GND	Signalmasse	GND	
22	GND	Signalmasse	GND	
23	GND	Signalmasse	GND	
24	GND	Signalmasse	GND	
25	GND	Signalmasse	GND	

Tabelle 2.1: Anschlußbelegung des parallelen Ports beim Amiga 500 und 2000 (weiblicher Stecker im Gerät)



Pin	Signal	Funktion	intern	Richtung
1	$\overline{\text{STROBE}}$	Strobe-Signal	8520-B $\overline{\text{PC}}$	aus
2	D0	Datenbit 0	8520-B PB0	ein/aus
3	D1	Datenbit 1	8520-B PB1	ein/aus
4	D2	Datenbit 2	8520-B PB2	ein/aus
5	D3	Datenbit 3	8520-B PB3	ein/aus
6	D4	Datenbit 4	8520-B PB4	ein/aus
7	D5	Datenbit 5	8520-B PB5	ein/aus
8	D6	Datenbit 6	8520-B PB6	ein/aus
9	D7	Datenbit 7	8520-B PB7	ein/aus
10	$\overline{\text{ACK}}$	Acknowledge	8520-B $\overline{\text{FLAG}}$	ein
11	BUSY	Drucker aktiv	8520-A PA0	ein/aus
			8520-A SP	ein/aus
12	POUT	Papierende	8520-A PA1	ein/aus
			8520-A CNT	ein/aus
13	SEL	Drucker on-line	8520-A PA2	ein/aus
14	GND	Signalmasse	GND	
15	GND	Signalmasse	GND	
16	GND	Signalmasse	GND	
17	GND	Signalmasse	GND	
18	GND	Signalmasse	GND	
19	GND	Signalmasse	GND	
20	GND	Signalmasse	GND	
21	GND	Signalmasse	GND	
22	GND	Signalmasse	GND	
23	+5V	Versorgungsspannung	+5V	
24	---	(nicht belegt)		
25	$\overline{\text{RESET}}$	Reset-Leitung	gepuffert	aus

Tabelle 2.2: Anschlußbelegung des parallelen Ports beim Amiga 1000 (männlicher Stecker im Gerät)

ein weiblicher, dann ist zu erkennen, daß sich die Belegung im wesentlichen vertauscht hat. Der Schaltplan aus Bild 2.1 beschreibt einen Adapter, mit dem Zusätze für den Amiga 1000 auch an den Amiga 500 bzw. 2000 angeschlossen werden können. Benötigt werden lediglich zwei männliche 25-polige Sub-D-Stecker, etwas Schaltlitze und viel Geduld. Achten Sie beim Zusammenlöten genau auf die in den Steckern eingeprägten Pinnumerierungen.

2.1.2 Spannung bitte

An der Steckverbindung liegt auch die Versorgungsspannung +5 Volt vom Netzteil. Das ist sehr gut für kleine externe Schaltungen. Allerdings erfolgt die Zuführung außer beim Amiga 1000 über einen Widerstand von 47 Ohm. Er wurde zur Sicherheit eingefügt, denn käufliche Druckerkabel verbinden oft jeden Pin mit dem korrespondierenden Anschluß am Drucker. Dort könnte aber Masse anliegen, so daß ein Kurzschluß entstehen würde. Die Folge wäre ein Systemabsturz oder gar ein Schaden am Netzteil. Der zwischengeschaltete Widerstand verhindert diesen Effekt. An ihm würde gegebenenfalls die gesamte Spannung abfallen. Dann fließt zwar ein Strom von gut 106 mA, doch die Funktion bleibt erhalten, und nichts wird beschädigt. Für uns heißt das leider, daß wir mit TTL-Schaltungen nur einen Strom von etwa 5 mA ziehen können, da andernfalls die minimale TTL-Versorgungsspannung unterschritten würde. Mit anderen Worten: Der +5-V-Anschluß ist so nicht brauchbar!

Die in diesem Kapitel vorgestellten Schaltungen sind trotzdem auf eine Versorgung über den Port-Stecker ausgelegt, denn das Netzteil selbst kann genügend Strom zur Versorgung kleiner externer Schaltungen liefern. Da Sie als Bastler höchstwahrscheinlich Ihre Kabel selber herstellen bzw. fertige entsprechend modifizieren werden, ist es empfehlenswert, den Sicherheitswiderstand im Rechner zu überbrücken. Sie finden ihn leicht, wenn Sie die Leiterbahn von Pin 14 ausgehend verfolgen.

Sollten Sie keinen Eingriff in den Rechner wagen, zum Beispiel weil die Garantie noch nicht abgelaufen ist, dann muß auf den Platinen jeweils die Verbindung zum +5V-Anschluß unterbrochen und zwischen dieser Stelle und Masse die Versorgungsspannung von einem externen Netzteil her eingespeist werden.

Ähnlich liegen die Verhältnisse leider auch bei der RS-232-Schnittstelle, die im Kapitel 3 zum Einsatz kommen soll. Dort wurden im Amiga 2000 sogar 470-Ohm-Widerstände in die Leitungen für +12 Volt und –12 Volt geschaltet. Auch hier empfiehlt sich die Überbrückung dieser Widerstände, wenn kein externes Netzteil verwendet werden soll.

Bild 2.2 zeigt die Schaltung eines geeigneten Netzteils für beide Fälle. Der Trafo setzt die Netzspannung auf 2 mal 15 Volt herab. Danach folgt ein Brückengleichrichter. Diese Bauteile gibt es fertig in Gehäusen mit vier Anschlüssen. Aufgedruckt sind die Maximalwerte für Spannung in Volt und Strom in Milliampere. Die Angabe B80/C1500 bedeutet zum Beispiel maximal 80 Volt und 1,5A.

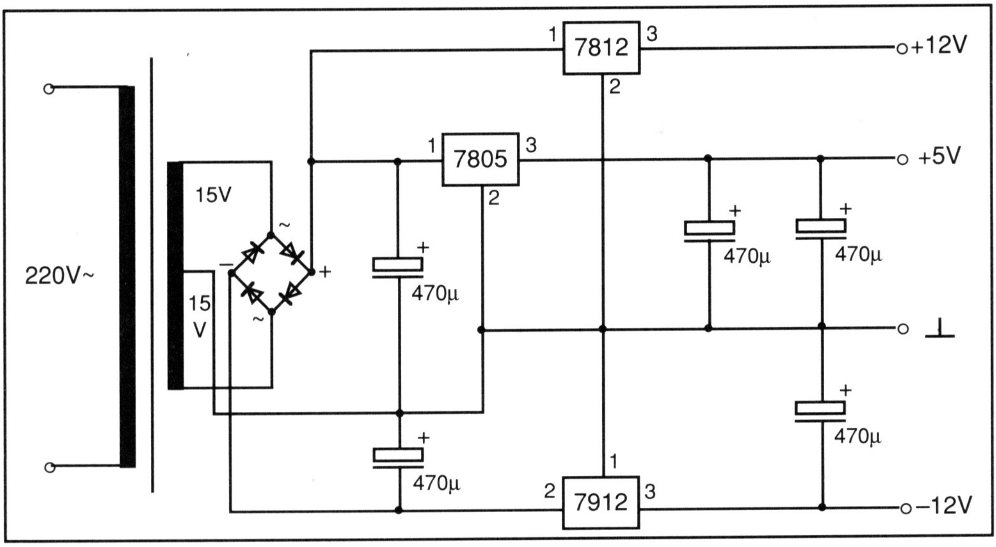


Bild 2.2: Ein Netzteil zur Versorgung der Zusatzschaltungen

Gerade digitale TTL-Bausteine arbeiten nur in einem engen Spannungsbereich zuverlässig. Erlaubt sind Schwankungen um ±5%, also liegen die Grenzen der Betriebsspannung bei +4,75 Volt und +5,25 Volt. Daher werden hier für jede Spannung elektronische Regler vom Typ 78XX (positiv) bzw. 79XX (negativ) eingesetzt. Bemerkenswert ist bei diesen Spannungsreglerserien der eingebaute doppelte Überlastungsschutz. Die Bausteine reagieren in zweifacher Weise auf Überlastung.

Bei plötzlich auftretenden kurzzeitigen Überströmen mindert und begrenzt die Innenschaltung den Strom auf einen erträglichen Wert. Bei thermischer Überlastung jedoch schaltet sie – ähnlich wie ein Sicherungsautomat – ganz ab. Damit ist das Netzteil ohne weitere Maßnahmen bereits kurzschlußfest. Standardregler vertragen einen Dauerstrom von 1A. Dazu müssen sie allerdings auf ein Kühlblech montiert werden. Daneben sind auch Ausführungen für diverse andere Maximallasten lieferbar, zum Beispiel 78LXX für 0,5A und 78SXX für 2A. Die Anschlußbelegung der Standard-Regler-ICs zeigt Bild 2.3.

Sowohl Ein- als auch Ausgang des Reglers sind mit Kondensatoren abgeblockt, um Störungen zu unterdrücken.

Damit die Schaltung überhaupt richtig arbeiten kann, muß der vorgeschaltete Transformator neben der richtigen Spannung auch genügend Strom zur Verfügung stellen. Andernfalls würde die Spannung bei größerer Last einfach zusammenbrechen. Für unseren Bedarf genügen etwa 800mA vollauf.

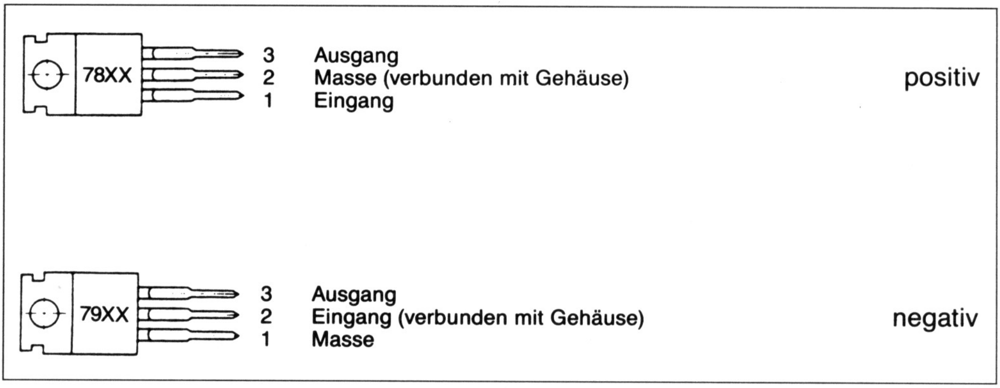


Bild 2.3: Positiv- und Negativspannungsregler 78XX und 79 XX

2.2 Safety First

Bevor Sie sich nun mit einem Lötkolben bewaffnen und gnadenlos Ihren Rechner traktieren, zunächst noch eine Warnung: Die Portbausteine sind empfindlich gegen grobe Behandlung und können durch Unachtsamkeit leicht zerstört werden. Ein Ersatzteil ist dann schwer zu bekommen und wahrscheinlich auch recht teuer. Lesen Sie also zunächst die folgenden Zeilen aufmerksam durch.

Voraussetzung für alle Experimente sind passende Stecker, an die Kabel angelötet werden können. Ein Zusatz darf nie aufgesteckt oder abgezogen werden, während der Amiga oder das Zusatzgerät eingeschaltet ist, da beim Potentialausgleich kurzfristig unzulässige Spannungen entstehen und empfindliche Bauteile zerstören könnten.

Ein zweiter Punkt kommt bei Schaltungen zum Tragen, die selbst mit ihrem Ausgang an den Port angeschlossen werden. Es muß genau darauf geachtet werden, daß nicht zwei Ausgänge gegeneinander arbeiten. Über kurz oder lang würde mindestens einer der Ausgangstransistoren zerstört werden. Unbenutzte Portanschlüsse sollten sicherheitshalber immer als Eingang programmiert werden. Beim Einschalten des Rechners (und bei jedem anderen Reset) wird unter anderem aus diesem Grund das gesamte Datenrichtungsregister automatisch gelöscht. Alle Portleitungen befinden sich so mit Sicherheit im Eingangsmodus, bis das Betriebssystem oder der Anwender sie je nach ihrer Bestimmung programmiert.

Eine weitere Regel betrifft die angeschlossene Schaltung selbst: Sollen externe Geräte oder hohe Spannungen vom Computer aus geschaltet werden, ist es unbedingt ratsam, den Weg über ein Relais oder einen Optokoppler zu wählen. So schützt man seine wertvolle Ausrüstung am wirkungsvollsten vor möglichen Schäden. Beide Schaltungsprinzipien trennen die Last galvanisch vom Portausgang, das

heißt, es existiert keine leitende Verbindung zwischen Computer und dem angeschlossenen Gerät. Damit braucht nicht ständig befürchtet zu werden, daß der maximal zulässige Schaltstrom überschritten und der Portbaustein zerstört wird, oder daß über ein angeschlossenes 220-Volt-Gerät eventuell sogar Netzspannung am System anliegt. Den Lastteil sollte man – sofern er irgendwie mit Netzspannung zu tun hat – ohnehin in ein gut isoliertes Gehäuse einbauen, auch wenn er »nur zum Ausprobieren« aufgebaut wurde. Jegliche Berührung mit Netzspannung führenden Teilen ist lebensgefährlich! Dessen sollte man sich ständig bewußt sein.

2.3 Datenausgabeschaltungen

Jetzt geht's endlich los. Im folgenden Abschnitt werden einige Grundschaltungen angegeben, über die in irgendeiner Weise die Umwelt des Computers beeinflußt werden kann.

2.3.1 Erste Hilfe

Wer ohne viel Aufwand testen will, ob das bisher über den Port Gesagte auch wirklich stimmt, kann einfach mit einem Vielfachmeßgerät die Spannungen an den Portanschlüssen nachmessen. Die meist schwarze Strippe für den Minuspol klemmt man dazu an einen der reichlich vorhandenen Masseanschlüsse GND (Ground). Die andere Strippe kommt an einen beliebigen Portanschluß PB0 bis PB7. Unter Basic schaltet man nun wie oben erläutert mit

```
POKE 12571392,255
```

alle Leitungen auf Ausgabe (12571392 dezimal = BFD300 hexadezimal, 255 dezimal = 11111111 dual), und legt dann zunächst mit

```
POKE 12570880,255
```

diese Ausgänge alle auf HIGH (12570880 dezimal = BFD100 hexadezimal), dann läßt sich am ausgewählten Meßpunkt eine Spannung knapp unter +5 Volt messen.

Nun bringt

```
POKE 12570880,0
```

alle Portausgänge wieder auf LOW (0 dezimal = 00000000 dual). Die Nadel des Meßgeräts schwingt zurück und zeigt jetzt einen kaum meßbaren Wert dicht bei 0 Volt.

Probieren Sie die letzten beiden POKE-Befehle mehrmals hintereinander aus. Der Zeiger muß jedesmal hin- und herschwingen. Das gleiche geschieht natürlich auch an allen anderen Portleitungen.

2.3.2 Pegel sichtbar – Datenausgabe über LED

Eine Leuchtdiodenanzeige der anliegenden Spannung ist sehr leicht zu realisieren. Allerdings sollte aus Sicherheitsgründen ein Transistor vorgeschaltet werden, um den zulässigen Schaltstrom des Portausgangs (maximal zwei TTL-Lasten) nicht zu überschreiten. Bild 2.4 zeigt die ganze Schaltung. Sie funktioniert folgendermaßen:

Die Leuchtdiode (Light Emitting Diode = LED) ist über einen Vorwiderstand und den Transistor zwischen +5 Volt und

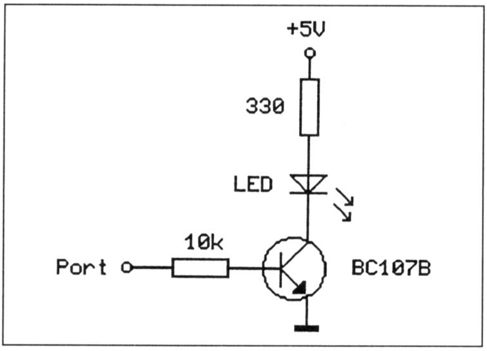


Bild 2.4: Ansteuerung einer Leuchtdiode

Masse geschaltet. Leuchtdioden müssen immer über einen Vorwiderstand betrieben werden, der den Strom begrenzt und damit gleichzeitig die Helligkeit einstellt.

Der Transistor wird hier als ein vom Basisanschluß steuerbarer Schalter betrieben und liegt ebenfalls über einen Basisvorwiderstand am Port. Führt der Portausgang HIGH-Spannung, so nimmt die Kollektor-Emitter-Strecke des Transistors einen sehr kleinen Widerstand an. Denkt man sich den Transistor als Schalter, so ist dieser geschlossen. Es fließt ein genügend großer Strom, um die LED zum Leuchten zu bringen.

Sinkt die Portspannung auf LOW, dann steigt der Widerstand des Transistors und es kann kaum noch Strom durch ihn fließen. Der gedachte Schalter ist jetzt geschlossen. Also erlischt die Leuchtdiode.

Bild 2.5 zeigt die Innenschaltung eines TTL-NAND-Gatters mit offenem Kollektor. Auch hier wird der Ausgang ganz ähnlich über einen Schalttransistor gesteuert.

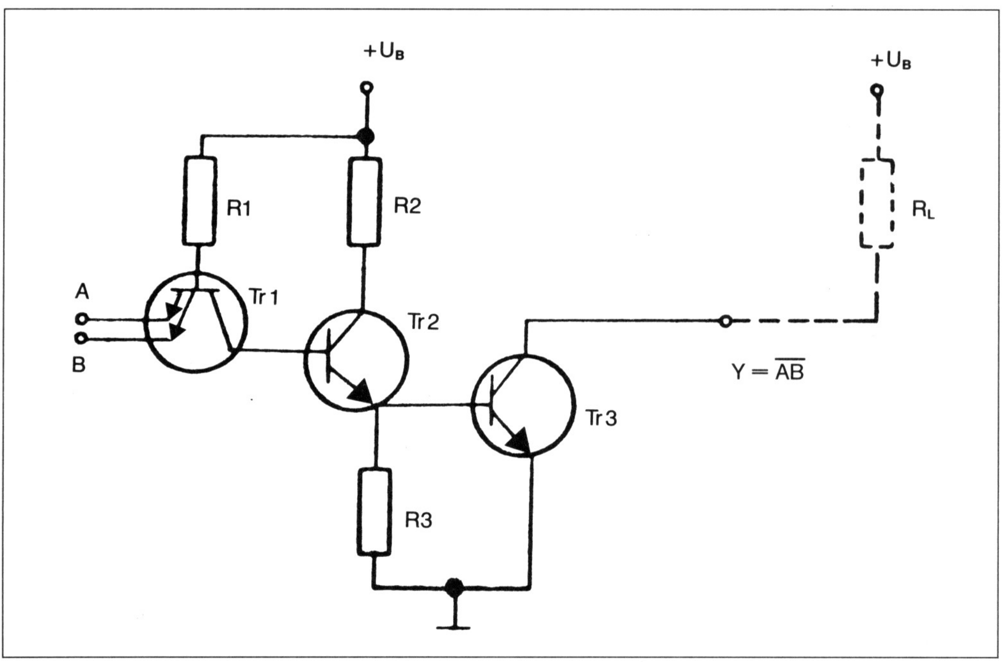


Bild 2.5: NAND-Gatter mit offenem Kollektor als Ausgang

Daher lassen sich statt diskreter Elektronik – also statt Transistoren und Widerständen – auch entsprechende ICs verwenden.

Sicherlich haben Sie schon bemerkt, daß

2.3.3 Krach machts

Nicht nur Leuchtdioden können von einem Schalttransistor gesteuert werden, sondern schlicht alles, was Strom braucht. Dazu gehört auch ein Piezo-Summer mit

Die Transistortypen hängen von der Größe des angeschlossenen Motors ab. Sie müssen die dabei entstehenden Ströme sicher schalten können. Geeignet sind etwa die komplementären Typen BD137/BD138.

2.3.5 Schrittmacher

Ein Schrittmotor wandelt elektrische Impulse in mechanische Drehbewegung mit definiertem Drehwinkel um. Er stellt ein mechanisches Bauelement dar, dessen Achse, den Steuerimpulsen folgend, schrittweise rotiert. Schrittmotoren werden überall dort eingesetzt, wo genau festgelegte und reproduzierbare Bewegungen nötig sind, beispielsweise als Antrieb für den Schreib-/Lesekopf in Floppylaufwerken oder des Druckkopfes in Druckern und Plottern, zur Blendensteuerung in Kameras, in Kurven-Lochstreifen- oder Kartenschreibern sowie in Robotern.

Jedesmal, wenn der Schrittmotor geeignet angesteuert wird, dreht sich seine Achse um einen durch die Bauart genau festgelegten Winkel. Übliche Schrittmotoren mit kleiner Schrittzahl drehen die Achse bei jedem Schritt um 7.5, 15, 45 oder 90 Grad. Motoren mit höherer Schrittzahl haben Standardschrittweiten von 1.8 und 5.0 Grad. Bei bekanntem Anfangspunkt der Bewegung läßt sich die Position der Achse zu jeder Zeit genau angeben. Um eine bekannte Anfangsposition zu erreichen, wird üblicherweise die Welle an einen Anschlag gefahren. Alle nachfolgenden Impulse in dieser Richtung haben nun keine Wirkung mehr, da sie die Mechanik nicht ausführen kann. Nach diesem Vorgang muß von der Position des Anschlags ausgegangen werden. Jedesmal beim Einschalten eines Druckers zum Beispiel findet eine solche Kalibrierung statt.

Der Schrittmotor selbst besitzt in der Regel vier Spulen, bezeichnet mit S1, S2, S3 und S4. Werden sie in geeigneter Reihenfolge mit Stromimpulsen angesteuert, führt der Motor einen Schritt aus. Dazu gibt es drei mögliche Verfahren:

- Wenig Leistungsbedarf
- Normalbetrieb
- Halbschrittverfahren

Bei der ersten Möglichkeit werden Stromimpulse der Reihe nach durch alle vier Spulen geschickt: zuerst durch S1, dann durch S2, durch S3, durch S4 und dann wieder durch S1 und so fort. Bild 2.8 macht das deutlich. Nie werden zwei Spu-

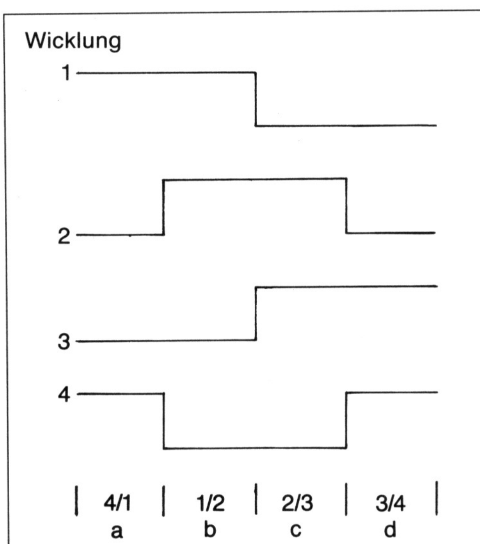


Bild 2.8: Zeitablauf der Wicklungsansteuerung im Normalverfahren

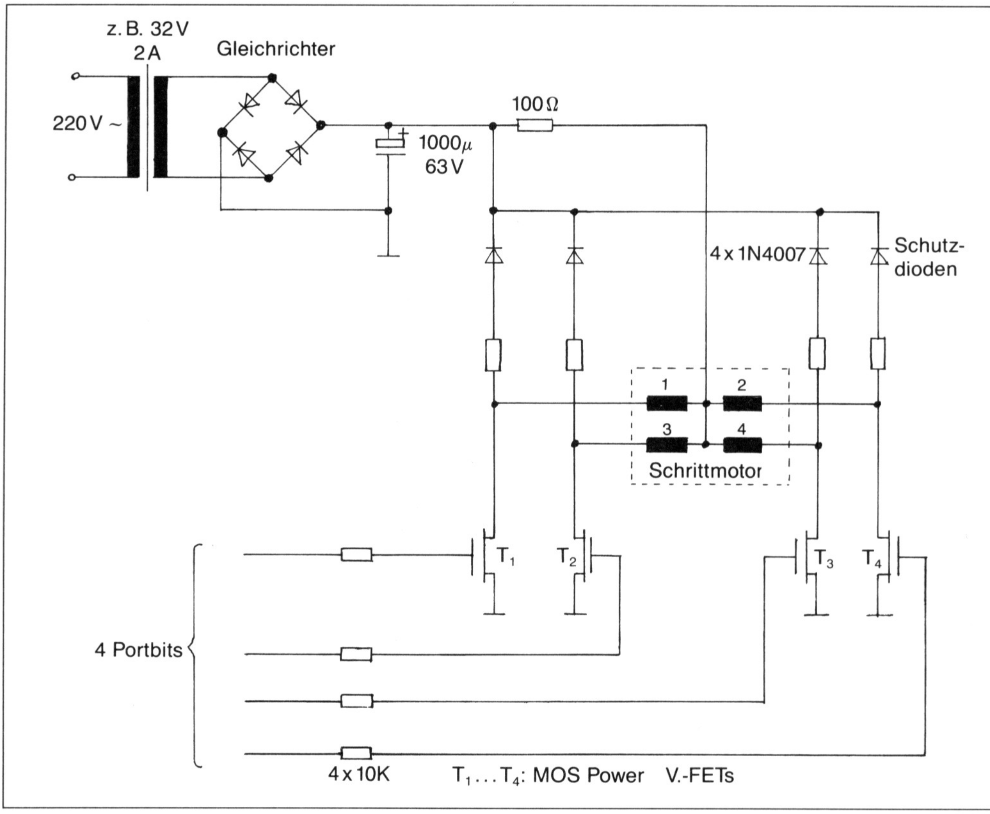


Bild 2.9: Ansteuerung eines Schrittmotors

len zur selben Zeit durchflossen und der Motor macht einen Schritt pro Impuls. Das Normalverfahren steuert immer zwei Wicklungen zugleich in folgender Reihenfolge an: S1 und S2, S2 und S3, S3 und S4, S4 und S1 und so weiter. Es ergibt sich eine sanftere Arbeitsweise des Motors, jedoch wird mehr Leistung benötigt.

Das Halbschrittverfahren erlaubt, zwischen jeden ganzen Schritt einen halben einzufügen. Die Ansteuer-Reihenfolge lautet hier: 1 und 2, 2, 2 und 3, 3, 3 und 4, 4, 4 und 1, 1 und so fort.

Die Spulen des Motors besitzen mit typisch 0,2 Ohm einen sehr kleinen Innenwiderstand bei recht hoher Induktivität. Daher sind besondere Entwurfstechniken für die Last-Transistoren und die Stromzuführung nötig, um die Schaltung vor Zerstörung durch die induzierten Spannungsspitzen zu schützen. Bild 2.9 zeigt eine Interfaceschaltung für Schrittmotorbetrieb.

Mit dem IC SAA 1027 von Siemens kann der Ansteuervorgang wesentlich erleichtert werden. Bild 2.10 zeigt die zugehörige

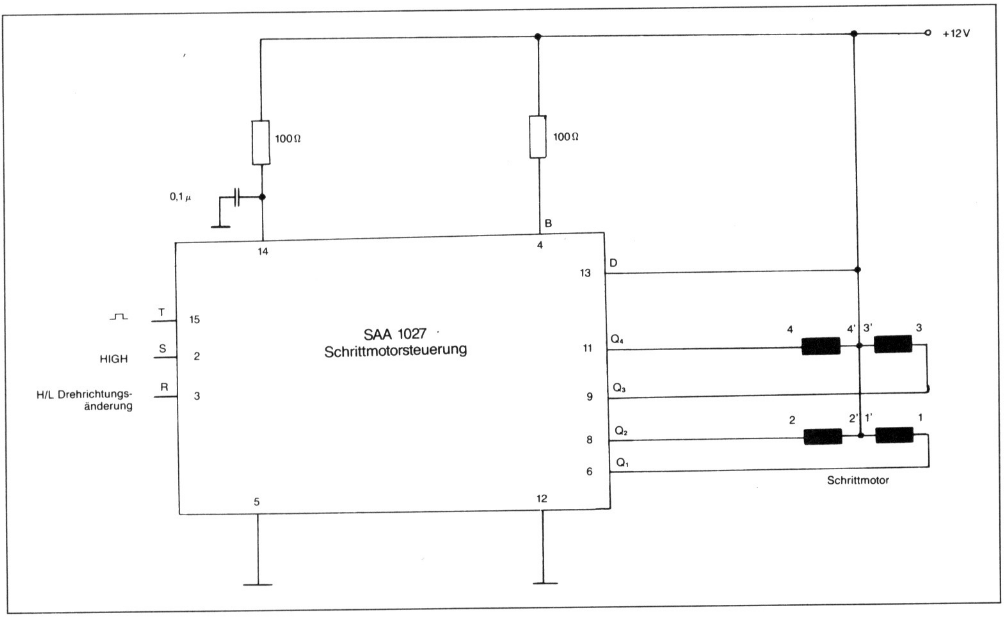


Bild 2.10: Schrittmotoransteuerung mit dem IC SAA 1027

Schaltung. Es ist nur noch ein Impulseingang nötig (Pin 15), über den mittels einer Frequenz die Drehgeschwindigkeit des Motors bestimmt wird. An Pin 3 legt HIGH- oder LOW-Pegel die Drehrichtung fest.

Natürlich dürfen die Impulse nicht zu schnell aufeinanderfolgen. Der Motor muß genügend Zeit haben, seine Achse zu drehen. Die maximal erreichbare Geschwindigkeit ist von Motor zu Motor verschieden. Damit verschiedene Bauformen verwendet werden können, wird meist ein gewisser Sicherheitsabstand eingehalten. So auch bei den Step-Impulsen zur Kopfbewegung in den Amiga-Diskettenlaufwerken. Verantwortlich dafür ist das Trackdisk-Device. Nach dem Einschalten beträgt die Steprate 3 ms. Das Programm TDChange gibt Ihnen die Möglichkeit, diese Zeit zu variieren. Es wird aufgerufen mit

```
TDChange DRIVE STEPDELAY SETTLEDELAY
```

DRIVE steht für die Laufwerksnummer, STEPDELAY gibt die Zeit zwischen den Schritten an, SETTLEDELAY die Kopfberuhigungszeit vom Erreichen der gewünschten Spur bis zum Beginn des Zugriffs, beide in Mikrosekunden.

```
TDChange DF0: 3000 15000
```

ist die Standardeinstellung. Je nach Laufwerk läßt sich STEPDELAY bis etwa 900, also 0,9 ms herabsetzen, SETTLEDELAY bis 7000. Der Diskettenzugriff wird dadurch merklich schneller. Allerdings kann eine zu kurze Kopfberuhigungszeit auch

mehrere Leseversuche nötig machen und damit die Gesamtzeit verlängern.

2.3.6 Galvanisch getrennt, doch funktionell vereint

Ebenso wie in der LED- und der Summer-Schaltung liegen die Verhältnisse auch bei der Relaisschaltung nach Bild 2.11. Der Transistor schaltet wieder den Strom. Dabei darf keinesfalls die Schutzdiode parallel zum Relais vergessen werden. Sie schließt negative Spannungsspitzen kurz, die – ähnlich wie beim Motor – durch Abfallen des Ankers im Inneren der Relaisspule aufgrund von Induktionswirkung entstehen und den Schalttransistor zerstören könnten.

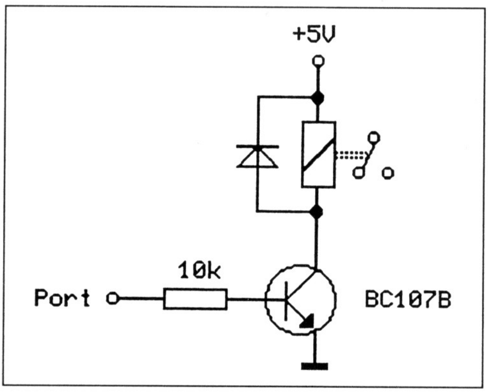


Bild 2.11: Schalten mit dem Relais am Computer

Durch die Spule des Relais erfolgt bei dieser Anordnung die Kopplung von Ursache und Wirkung nicht galvanisch, das heißt elektrisch leitend, sondern über ein magnetisches Feld. Sobald Strom durch die Wicklung fließt, zieht der Anker an und der Lastkontakt wird geschlossen.

Für größere Lastströme, wie sie zum Beispiel in Glühlämpchen oder stärkeren Relais für höhere Schaltleistungen leicht auftreten können, muß die Anordnung nach Bild 2.12 aufgebaut werden. Auch hier – wie immer beim Einsatz von Spulen und bewegten Teilen – die Schutzdiode bitte nicht vergessen!

Über die Relaiskontakte ist es nun möglich, die unterschiedlichsten Geräte zu schalten. Dabei ist der Lastkreis (Verbraucher) galvanisch völlig unabhängig vom Steuerkreis (Computer), so daß ohne weiteres auch Wechselströme schaltbar sind.

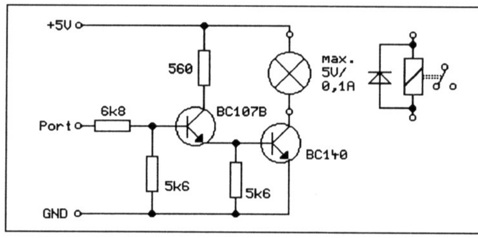


Bild 2.12: Galvanisch gekoppelte Schaltung für größere Lasten, zum Beispiel starke Relais (anstelle der Glühbirne)

2.3.7 Die Sache mit dem Optokoppler

Eine andere Realisierungsmöglichkeit der galvanischen Trennung ist die Kopplung mittels Licht. Bild 2.13 zeigt strenggenommen zwei Schaltungen. Links ein gewöhnlicher Anzeigekreis mit einer Leuchtdiode nach Bild 2.4. Die LED leuchtet bei HIGH-Pegel und erlischt bei LOW am Steueranschluß. Dicht neben dieser Leuchtdiode befindet sich ein Fototransistor, der über einen Thyristor zum Beispiel Netzspannung schaltet. Die Gleichrichterbrücke ermöglicht eine Ausnutzung aller Halbwellen des Wechselstroms.

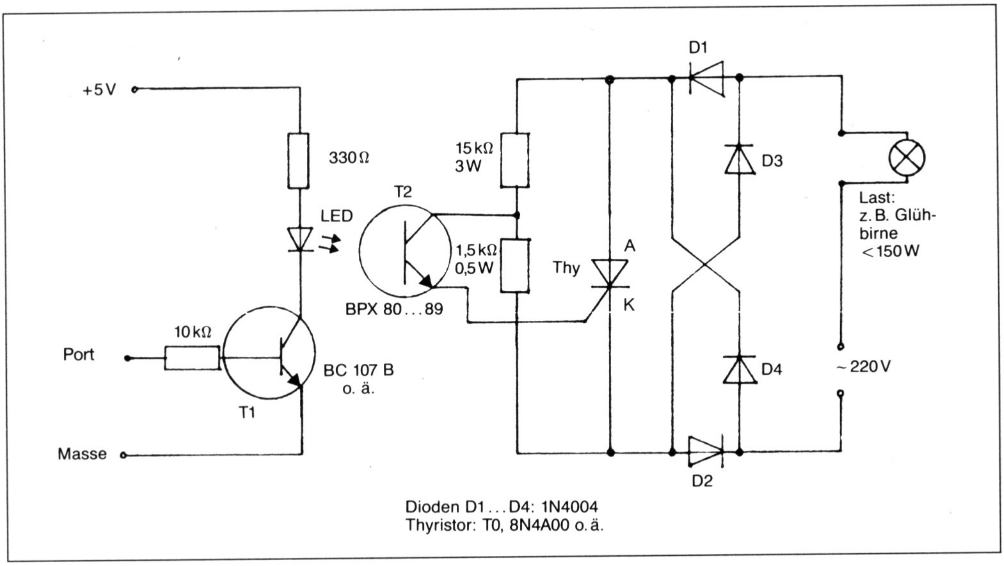


Bild 2.13: Schalten einer 220-Volt-Last, galvanisch vom Computer getrennt ohne Relais (optische Kopplung)

Solche und ähnliche kontaktlose, gleichspannungsgesteuerte Wechselstromschaltstufen finden bei Lichtsteuerungen, Heizungsregelungen oder Motorsteuerungen häufig Verwendung.

Die Anordnung Leuchtdiode – Fototransistor gibt es auch fertig als Optokoppler, lichtdicht eingegossen in ein gemeinsames Gehäuse. Die Anschlußbelegung geeigneter Typen finden Sie in Bild 2.14, die des verwendeten Thyristors in Bild 2.15.

Ein Thyristor hat die Eigenart, immer nur eine Halbwelle durchzulassen. Daher werden Doppelthyristoren – sogenannte Triacs – angeboten, die diesen Nachteil beheben. Bild 2.16 zeigt eine entsprechen-

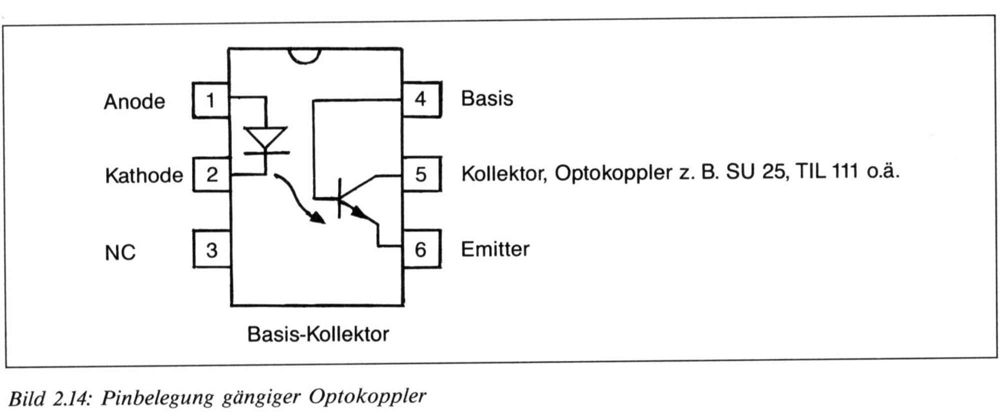
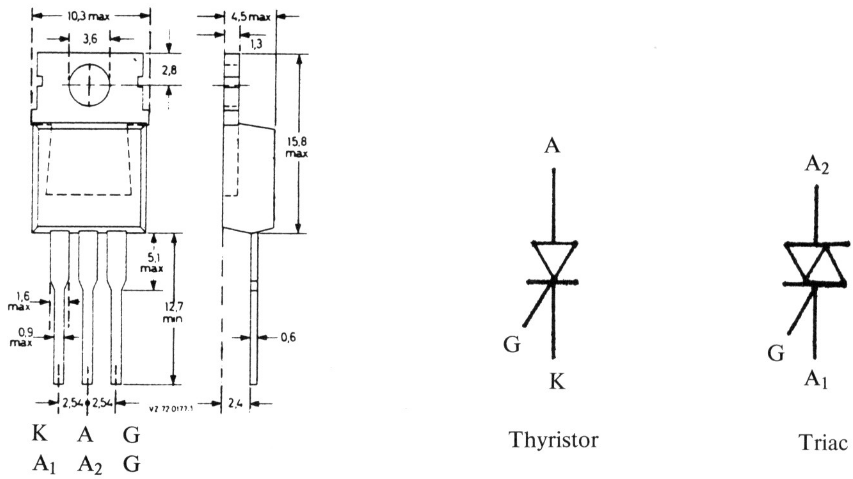


Bild 2.14: Pinbelegung gängiger Optokoppler



bei Thyristoren z. B. TYN 058...1008 (8 A)
bei Triacs z. B. TIC 226 D (8 A)

Bild 2.15: Schaltsymbole von Thyristor und Triac mit Pinbelegung im Gehäuse TO 220

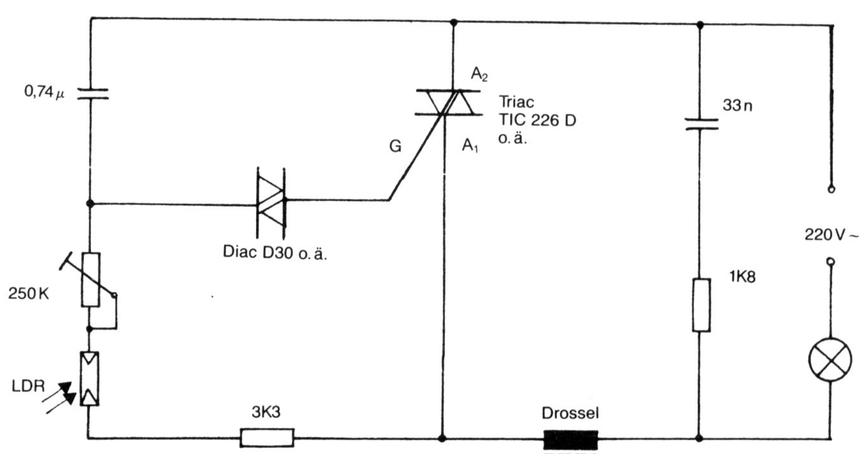


Bild 2.16: Schaltung zum Schalten von Glühbirnen

de Schaltung, durch die beide Halbwellen ausgenutzt werden. Zusätzlich sorgen einige passive Bauelemente für eine wirkungsvolle Entstörung. Auch hier erfolgt die Ansteuerung mittels Licht. Die Anschlußbelegung von Triacs ist ebenfalls in Bild 2.15 enthalten.

Natürlich lassen sich auch andere galvanisch trennende Elemente einsetzen, zum Beispiel das elektronische Lastrelais Nr. 18 66 35-46 von Conrad Electronic, Hirschau. Bei äußerst kompakter Bauart enthält es Optokoppler, Lastteil und Entstörglieder in einem Gehäuse.

2.3.8 Warum nicht auch mal regeln?

Die zuletzt beschriebenen Anordnungen bieten gegenüber den Relaisschaltungen neben der entfallenden mechanischen Kontaktbelastung vor allem den Vorteil größerer Schaltgeschwindigkeit. Damit ergeben sich vollkommen neue Möglichkeiten:

Schaltet man den Ansteueranschluß mit dem Computer sehr schnell ein und aus, kann man über die Schaltfrequenz eine angeschlossene Glühbirne beispielsweise rechnergesteuert dimmen, das heißt in ihrer Helligkeit regeln.

Das Zeitdiagramm in Bild 2.17 veranschaulicht diese Betriebsart für die Thyristor-Schaltung: Sobald die Steuerspannung am Port HIGH-Pegel erreicht, leuchtet die LED auf, und der Thyristor schaltet sofort durch: Im Lastkreis kann Strom fließen. Anders beim Umschalten der Port-Spannung auf LOW. Die Leuchtdiode erlischt zwar sofort, eine Eigenschaft des Thyristors ist es aber, erst ab dem nächsten Nulldurchgang der Netzspannung wieder zu sperren. Die Last wird in Wirklichkeit mit Wellenpaketen betrieben.

Falls die Schaltpausen kurz genug gewählt sind, merkt man das bei den meisten Verbrauchern nach außen hin nicht, da zum Beispiel der Glühfaden in einer Lampe immer noch eine gewisse Zeit nachleuchtet und damit die zu erwartende Flackerwirkung kompensiert.

Denkbar wäre als Steuerprogramm eine Interruptroutine, die bei jedem Aufruf

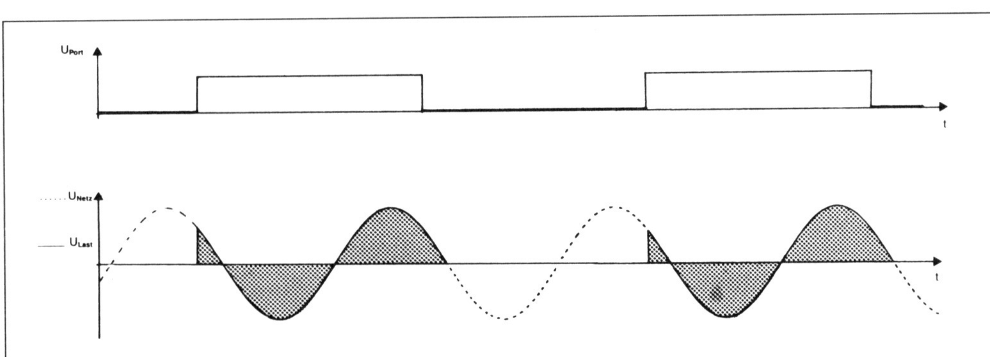


Bild 2.17: Zeitdiagramm zum Schaltverhalten der Anordnung aus Bild 2.9

den Portanschluß triggert, also zunächst einschaltet und gleich wieder ausschaltet. Der Lastkreis bleibt dann bis zum nächsten Nulldurchgang eingeschaltet. Wenn ein Timer diesen Interrupt auslöst, kann über den Zählerinhalt die Helligkeit gesteuert werden, während der Rechner wie gewöhnlich für andere Aufgaben zur Verfügung steht. Schließt man an mehrere Kanäle verschiedenfarbige Lampen an, können herrliche Lichtspielereien mit weichen Übergängen erzielt werden.

Ein lohnendes Einsatzgebiet für Fotofreunde wäre die Ansteuerung zweier (oder noch mehr) Diaprojektoren, die gemeinsam eine vom Rechner gesteuerte Show mit allen Schikanen absolvieren könnten, während der Amiga synchron die passende Musik liefert. Dazu müßte je ein Kanal in der beschriebenen Weise für die Dimmerung der Projektorlampen ausgebaut werden, ein anderer mit der Relaisschaltung nach Bild 2.11 für die Steuerung des Transports.

2.3.9 Was tun, wenn's stört?

Gleich noch ein Tip: Sollten Sie Probleme mit dem »Einschaltknacks« der gesteuerten Geräte haben, wenn also Ihr Computer beim Schalten von großen Lasten manchmal »aussteigt«, dann hilft folgende Entstörmaßnahme:

Ein Netzfilter nach Bild 2.18 wird in die Zuleitung des angeschlossenen Geräts gebracht. Jetzt können kurzzeitige Spannungseinbrüche über dessen Versorgungsleitung nicht mehr zum Netz gelangen und den Computer aus dem Tritt bringen. Als Drosselspulen L1...L4 eignen sich beispielsweise die Typen SFT 1030. Auch fertige Netzfilter – ausgebildet etwa als Kaltgerätestecker – werden angeboten. Leider sind sie meist nicht gerade billig. Bild 2.19 zeigt die Innenschaltung eines solchen Filters. Seine Funktionsweise beruht auf der Tatsache, daß der Innenwiderstand einer Spule mit steigender Frequenz zunimmt. Dagegen stellt ein

Bild 2.18: Entstörglied gegen Schaltspitzen

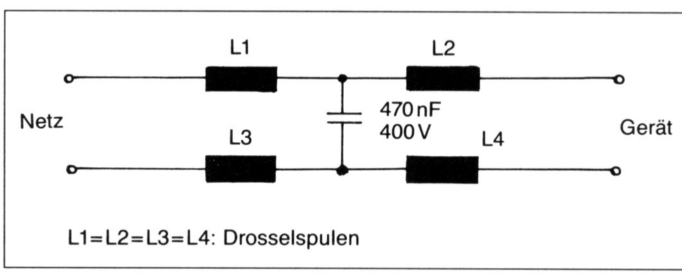
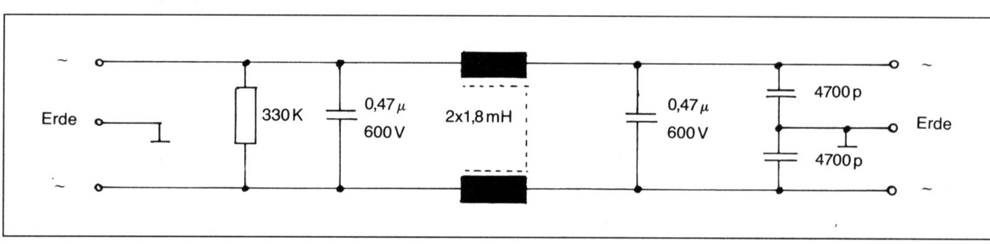


Bild 2.19: Schaltung eines Netzfilters



Kondensator für hochfrequente Störimpulse praktisch einen Kurzschluß dar.

Eine Unterdrückung von Störungen im Netz durch impulsweiser Trafo-Belastung kann eventuell bereits durch einen kleinen Widerstand von etwa 0,5 bis 1 Ohm behoben werden, der in Serie zur Sekundärwicklung des Trafos geschaltet wird. Als einfachste Möglichkeit sollten Sie zuvor aber einfach mal probieren, für das geschaltete Gerät eine weit entfernte Steckdose zu finden, über die Ihr Computer nicht gestört wird.

2.4 Dateneingabeschaltungen

Nun folgen einige Schaltungen zur Erfassung äußerer Zustände. Auch sie werden in ihrer Grundversion an jeweils eine Portleitung angeschlossen. War aber bei den Ausgabeschaltungen noch eine falsche Programmierung des Datenrichtungsregisters ungefährlich, so muß ab jetzt peinlich genau darauf geachtet werden, daß jede Portleitung, an der eine der folgenden Eingabeschaltungen betrieben wird, unbedingt als Eingang programmiert ist. Andernfalls kommt es zur Datenkollision, das heißt, falls der Computer und der angeschlossene Sensor unterschiedliche Spannungspegel auf ein und dieselbe Leitung legen, kann der entsprechende Ausgangstreiber im I/O-Baustein des Amiga zerstört werden!

2.4.1 Der Rechner streckt die Fühler aus

Es ist natürlich möglich, jede beliebige TTL-Schaltung an die Porteingänge anzu-

Meßgröße	geeigneter Wandler
Helligkeit	Fotodiode, Fototransistor, Fotowiderstand
Temperatur	Heißleiter (NTC = Negative Temperature Coefficient)
	Kaltleiter (PTC = Positive Temperature Coefficient)
	Bimetallschalter
Spannung	Analog/Digital-Wandler
Strom	Zurückführung auf Spannungsmessung durch Widerstand
Schall	Dynamisches, Elektret-Kondensator- oder Piezokristallmikrofon
Wegstrecke	Dehnungsmeßstreifen (bis 1mm), optische Abtastung, Ultraschall-Laufzeitmessung
Drehzahl	Tachogenerator, magnetischer Sensor, optische Abtastung, Nockenkontakt
Magnetfeld	Hall-Generator, magnetfeldabhängiger Widerstand
Luftfeuchte	Kapazitiver Feuchtigkeitssensor
Gasdruck	Silizium-Brückensensor, Druckmeßdose

Tabelle 2.3: Wandler zur Erfassung von Meßgrößen

schließen, da auch sie mit TTL-Pegeln arbeiten. Wenn diese Schaltung jedoch auf Informationen aus der Umwelt reagieren soll, muß sie mit geeigneten Wandlern bestückt sein. Oft ist eine sinnvolle Wandlung nur über größere Schaltungen möglich, da ihrer Natur nach analoge (stufenlose) Größen auf irgendeine Weise in digitale HIGH-LOW-Beziehungen umgesetzt werden müssen. Diese Schaltungen benötigen entsprechende Sensoren und nicht selten auch tiefergehende Kenntnisse aus der Analogtechnik. Tabelle 2.3 zeigt eine Auswahl von meßbaren Größen mit den zugehörigen Erfassungsmöglichkeiten.

2.4.2 Einfacher geht's nicht

Fangen wir beim Einfachsten an. Bild 2.20 zeigt, wie ein Porteingang per Hand zwischen HIGH und LOW umgeschaltet werden kann. So simpel die Schaltung aussieht, bildet sie doch die Grundlage für alle anderen Sensoren. Sie wird daher im folgenden ausführlich erläutert.

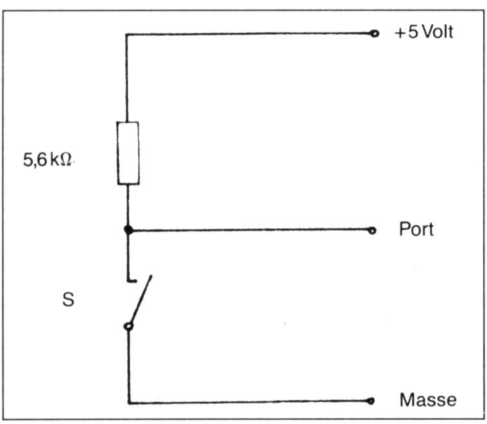


Bild 2.20: Einfachste Dateneingabe über einen Schalter

Ist der Schalter offen, liegt die Betriebsspannung von +5 Volt über den sogenannten »Pull-up-Widerstand« am Port. Weil nur ein sehr geringer Strom fließt, fällt am Widerstand so gut wie keine Spannung ab.

Schließt man den Schalter, so liegt der Porteingang direkt auf Massepotential. Am Widerstand fällt zwar die gesamte Betriebsspannung ab, doch der fließende Strom wird durch den verhältnismäßig großen Wert von 5600 Ohm ausreichend begrenzt. Dabei ist der Wert des Pull-up-Widerstands unkritisch. Er kann etwa zwischen 500 Ohm und 10 Kiloohm liegen.

Oft ist es wichtig, daß pro Tastendruck nur ein Impuls ausgegeben wird. Die Kontakte praktisch aller Taster prellen aber und geben dadurch jeweils in der Umschaltphase unkontrollierte Impulse ab. In Bild 2.21 oben ist ein solcher gestörter Spannungsverlauf dargestellt, darunter der gewünschte Verlauf. Um ihn zu erhalten, muß eine kleine Zusatzschaltung vorgesehen werden, beispielsweise die Anordnung aus Bild 2.22. Als Taster T ist jetzt eine Ausführung mit Umschalt-Kontakten einzusetzen. Schon beim ersten Berühren des jeweiligen Basis-Anschlusses wird die bistabile Schaltung umgesteuert und bleibt in diesem Zustand, bis die Basis des anderen Transistors wieder auf Masse-Potential gelegt wird.

Einzelne Taster werden durch das Keyboard des Computers keineswegs überflüssig gemacht, wie es auf den ersten Blick scheinen mag, denn in vielen von Computern gesteuerten Geräten muß ja

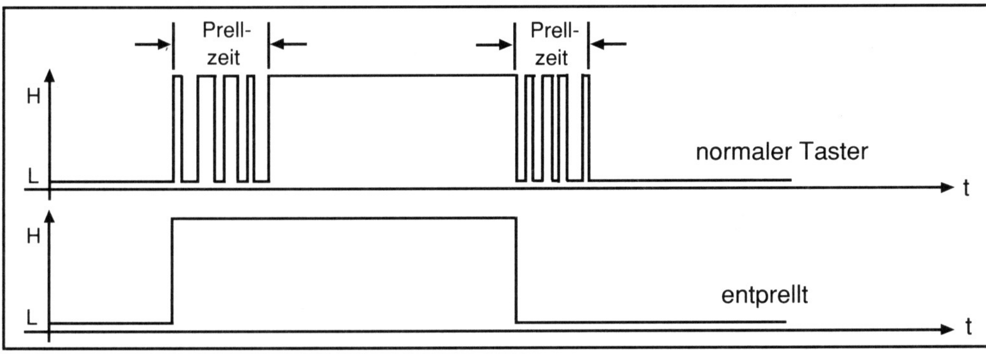


Bild 2.21: Spannungsverläufe beim Prellen eines Tasters

auf irgendeine Weise die Position von Hebeln, Greifern o.ä. an den Rechner zurückgemeldet werden. Anwendungen wären beispielsweise Endanschlagsmelder mit Springkontakten. Statt des Tasters können in anderen Fällen auch Quecksilber- oder Bimetallschalter eingesetzt werden.

2.4.3 Schalten per Licht

Die Eingabeschaltung nach Bild 2.20 gibt bereits eine Grundschaltung der digitalen Datenerfassung an. Das gleiche Prinzip mit Spannungsteiler taucht immer wieder auf. In der Anordnung nach Bild 2.23 wird bei genauerem Hinsehen der Schalter

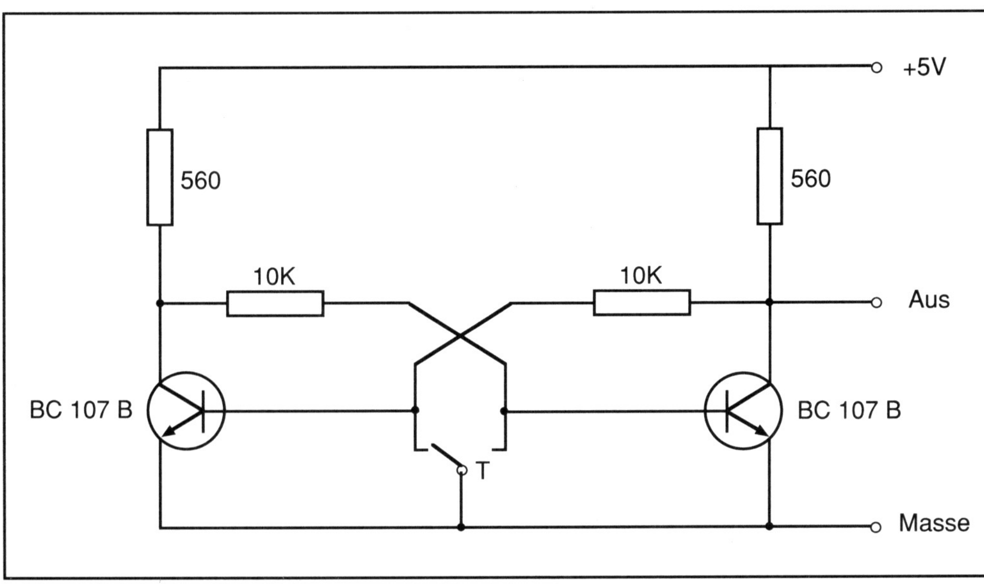


Bild 2.22: Schaltung zur Entprellung eines Tasters

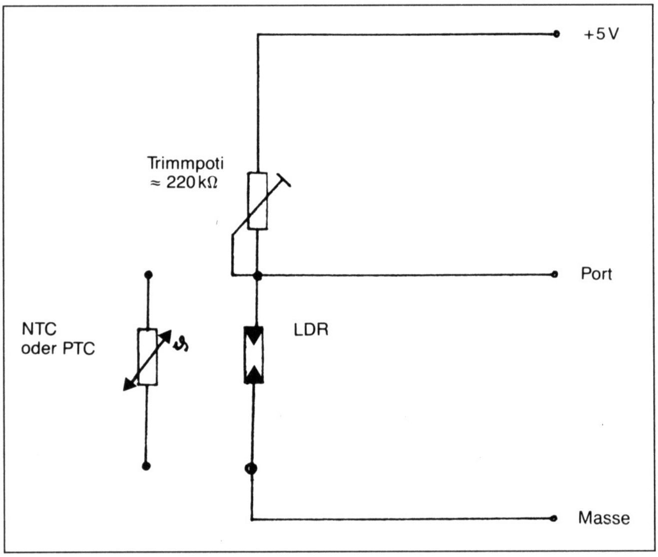


Bild 2.23: Licht oder Temperatur steuert den Computer

eigentlich nur durch den lichtabhängigen Widerstand (LDR) bzw. durch die temperaturabhängigen Meßfühler NTC oder PTC ersetzt. Die beiden Bauelemente bilden in jedem Fall einen Spannungsteiler, der ähnlich funktioniert wie eben schon erläutert. Der Pull-up-Widerstand wurde diesmal einstellbar gemacht, um die Empfindlichkeit regeln zu können.

Die angegebene Schaltung kann prinzipiell als Dämmerungsschalter benutzt werden, zum Beispiel um beim Unterschreiten einer gewissen Helligkeit die Beleuchtung einschalten zu lassen. Es empfiehlt sich, einen Baustein vorzuschalten, der am Ausgang schlagartig durchschaltet, wenn am Eingang eine bestimmte Schwelle unter- bzw. überschritten wird. Genau dieses Verhalten zeigt der Schmitt-Trigger, eine bestimmte Schaltvariante, die auch in einzelnen Bausteinen der 74XX-Reihe zu finden ist, beispielsweise im 7413 mit zwei NAND-Gattern zu je vier Eingängen. Bild 2.24 zeigt den Verlauf der Ausgangsspannung in Abhängigkeit vom Eingang. Man erkennt, daß der Ausgang bei einer etwas höheren Spannung von LOW nach HIGH schaltet, als im umgekehrten Fall von HIGH nach LOW. Der Unterschied zwischen den beiden Spannungen beträgt hier etwa 0,8 Volt. Er wird Hysterese genannt

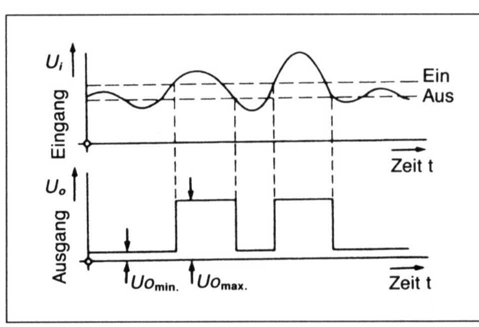


Bild 2.24: Verhalten einer Schmitt-Trigger-Schaltung

und verhindert Störungen, wie zum Beispiel ständiges Flackern in einem gewissen Übergangsbereich beim Dämmerungsschalter.

Mit einem LDR, einem Widerstand und beispielsweise einer Taschenlampe, ist schon der Aufbau einer Lichtschranke möglich. Dabei tritt eine Änderung des Ausgangszustandes auf, wenn die Lichtschranke unterbrochen wird.

2.4.4 Ein Lichtschranken-Modul

Lichtabhängige Widerstände sind billig und universell einsetzbar. Sie haben jedoch einen gravierenden Nachteil: Sobald die Lichtverhältnisse sehr schnell wechseln, kommen sie mit ihrer Widerstandsänderung einfach nicht mehr nach. Sie reagieren zu träge. Kurze Dunkelphasen werden verschluckt und mit der Empfindlichkeit steht es auch nicht gerade zum besten. In kritischen Fällen verwendet man daher gewöhnlich Fotodioden oder Fototransistoren.

Die Schaltung nach Bild 2.25 läßt sich als Lichtschrankensensor für schnelle Vorgänge einsetzen.

Sie benutzt einen preiswerten Fototransistor, der um so besser leitet, je mehr Licht auf ihn fällt. Er bildet zusammen mit seinem Vorwiderstand und dem Trimmpoti einen Spannungsteiler, der im beleuchteten Zustand so eingestellt wird, daß der folgende Schalttransistor T1 gerade leitet. An seinem Kollektor liegen daher praktisch 0 Volt, so daß T2 gesperrt ist und am Ausgang HIGH anliegt.

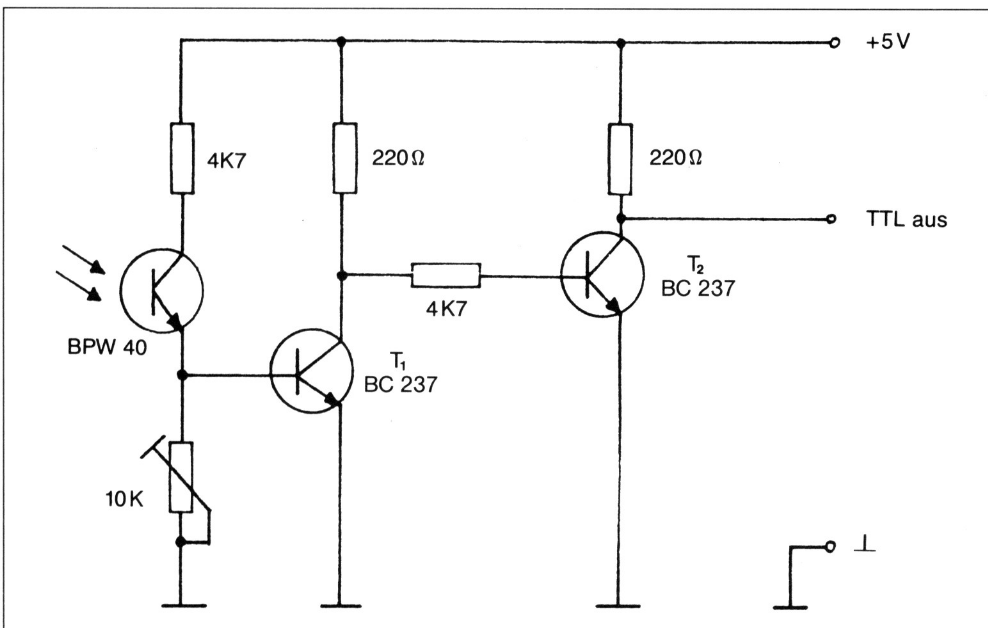


Bild 2.25: Schnell reagierender Lichtschrankenempfänger

Beim Abdunkeln der Beleuchtung erhöht sich der Innenwiderstand des Fototransistors. Die Basisspannung des Transistors T1 sinkt also, und am Kollektor von T1 steht nun relativ hohe Spannung. Damit schaltet T2 durch und der Ausgang wird LOW. Sein Zustand entspricht damit der Helligkeit am Fototransistor: bei Hell führt er HIGH, bei Dunkel LOW.

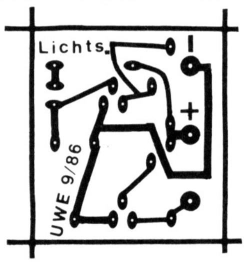


Bild 2.26: Layout des Lichtschrankenmoduls

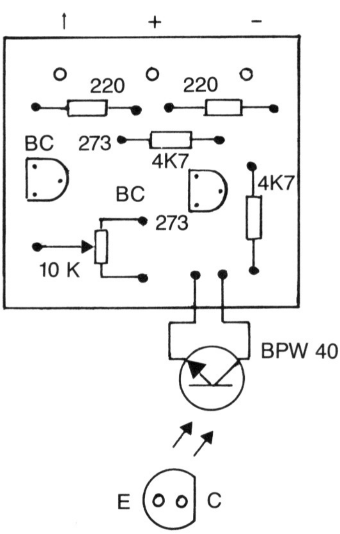


Bild 2.27: Bestückungsplan für das Lichtschrankenmodul

Da eine schnelle Lichtschranke bei vielerlei Anwendungen gute Dienste leisten kann, soll die Schaltung hier als kleiner Modul realisiert werden. Sie findet auf der Mini-Platine nach Bild 2.26 Platz, die gleichzeitig eine Befestigung für den Fototransistor darstellt. Das Modul ist an viele Auswerteschaltungen über Steckverbindungen direkt anschließbar. Dazu dienen die drei Lötnägel für +5 Volt, Masse und Signal. Die Bauteile (Tabelle 2.4) werden nach dem Bestückungsplan (Bild 2.27) verlötet.

1	Fototransistor	BPW 40
2	Transistoren	BC238B
2	Widerstände	220 Ohm
2	Widerstände	4,7 Kiloohm
1	Trimmpoti	10 Kiloohm
3	Lötnägel	
1	einseitige Leiterplatte nach Bild 2.26	

Tabelle 2.4: Die Bauteile für das Lichtschrankenmodul

2.4.5 Dateneingabe durch Handauflegen

Mit der Schaltung nach Bild 2.28 ist die Dateneingabe über Sensortasten möglich. Drei hintereinandergeschaltete Transistoren besitzen eine so große Verstärkung, daß bereits über den Hautwiderstand zwischen den beiden offenen Kontakten am Sensortaster S der Schaltvorgang ausgelöst wird. Die Verbindung der Sensortaste zur Betriebsspannung +5 Volt kann man auch weglassen, da der menschliche Körper als Antenne wirkt und aufgrund der überall vorhandenen Felder von Stark-

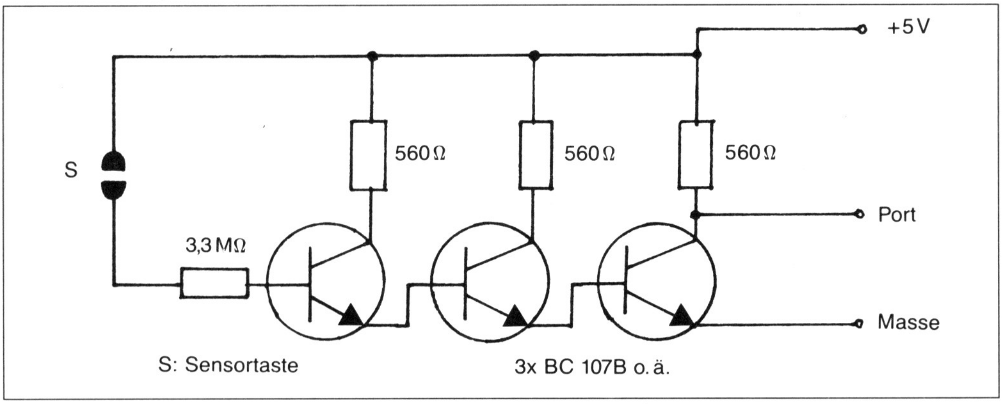


Bild 2.28: Dateneingabeschaltung über Sensortaste

stromnetzen bei Berührung eine ausreichend große Brummspannung an die Basis des ersten Transistors abgibt. Der hohe Basisvorwiderstand dient als Strombegrenzer für den Fall, daß die Sensortaste irrtümlich einmal niederohmig – zum Beispiel durch eine leitende Verbindung – überbrückt wird.

2.4.6 Der Computer bekommt Ohren

Einen Leckerbissen besonderer Art zeigt Bild 2.29. Mit dem Mikrofon kann der Computer sogar auf Schall reagieren. Zunächst bietet die Anordnung sicherlich einen für passionierte Elektroniker ungewöhnlichen Anblick: Da ist ein Inverter als Verstärker zweckentfremdet! Über den 1,7 Megaohm-Widerstand wird dazu sein Ausgang auf den Eingang zurückgekoppelt, so daß sich ein Arbeitspunkt knapp unterhalb der Umschaltspannung einstellt. Ähnlich wie bei einem Operationsverstärker ist dieser Widerstand übrigens für die Empfindlichkeit der Schaltung verantwortlich. Vergrößert man ihn, führen auch leisere Geräusche zur Auslösung.

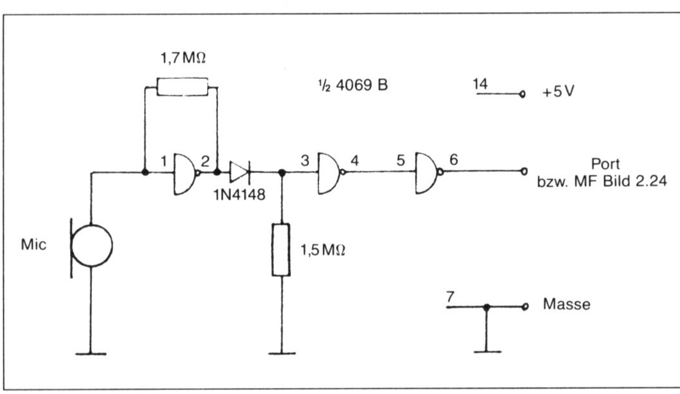


Bild 2.29: Steuerung durch Schall

Natürlich wächst damit aber die Störanfälligkeit durch unbeabsichtigte Umwelteinflüsse.

Bei Geräuschen kommt vom Mikrofon eine Spannung, die ausreicht, um am Ausgang ein starkes Signal zu erzeugen. Die positiven Spitzen werden über die Diode an die nachfolgenden Gatter weitergegeben, die sie als eindeutige logische Informationen an den Port durchschalten. Als Schallaufnehmer muß unbedingt ein Kristallmikrofon verwendet werden, da eine andere, niederohmige Signalquelle den Eingang sofort auf eine zu kleine Spannung herabziehen würde.

Einen Nachteil hat die Schaltung aber noch: Bei manchen Geräuschen sind die Ausgangsimpulse sehr schmal und könnten unter Umständen, wenn das Programm nicht gerade zur richtigen Zeit den Port abfragt, unerkannt verlorengehen. Falls beispielsweise ein Telefon klingelt, entsteht bei jedem Anschlagen des Klöppels an die Glocke ein kurzer Impuls an Pin 6 des Inverterbausteins. Das kann der Computer so nicht auswerten. Mit Hilfe eines nachgeschalteten Monoflops aus Bild 2.30 aber, wird daraus für jedes Klingeln ein einziger Impuls geformt, denn beim 74123 handelt es sich um ein retriggerbares Monoflop. Das heißt, beim ersten LOW-HIGH-Übergang am Eingang geht das Ausgangssignal für eine bestimmte, mit R und C einstellbare Zeitspanne auf HIGH. Erfolgt während dieser Zeit ein weiterer LOW-HIGH-Übergang, dann verlängert sich die Ausgangs-HIGH-Zeit des Monoflops entsprechend.

In unserem Beispiel ist diese Ausgangszeit mit der RC-Kombination auf etwa 0,6 Sekunden eingestellt. Allgemein errechnet sich die Impulslänge für LS-Typen zu

$$T_W = 0{,}45 * R * C.$$

Dabei müssen R in Ohm und C in Farad angegeben werden; die Zeit T_W ergibt sich dann in Sekunden.

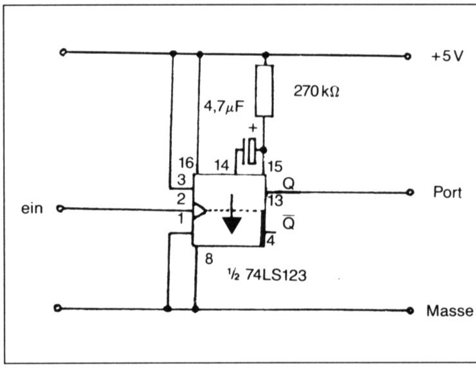


Bild 2.30: Impulsverlängerer

Anstelle des Mikrofons kann in die Schaltung nach Bild 2.29 auch ein Piezoelement eingesetzt werden, wie es für Alarmanlagen als Glasbruchmelder an Fensterscheiben verwendet wird. Ähnlich dem Kristallmikrofon gibt dieser Sensor bei plötzlicher mechanischer Belastung einen geringen Strom ab, der bereits zum Durchschalten der Gatter führt.

2.4.7 Amiga hört Radio

Als Signalquelle für die folgende Schaltung kann ein Kofferradio, Tonbandgerät, Plattenspieler oder eine ganze Stereoanlage dienen, egal ob mit Netz- oder Batteriebetrieb. Die Einsatzmöglichkeiten sind vielseitig und reichen von einer digitalen

Computer-Lichtorgel bis zu einer einfachen Spracherkennung.

Die Ankopplung nach Bild 2.31 ist so gewählt, daß der Verstärker der Quelle nur minimal belastet wird. Vom Lautsprecherausgang – oder vom eingebauten Lautsprecher direkt – wird die Tonfrequenz (NF) einem Übertrager mit dem Übersetzungsverhältnis 1:10 zugeführt, der die Aufgabe hat, den niederohmigen Ausgang des Verstärkers an den hochohmigen Eingang der Zusatzschaltung anzupassen, und das Signalgerät galvanisch von der Lichtorgelschaltung zu trennen. Am Ausgang des Übertragers (hochohmige Seite) liegt ein als Spannungsteiler geschaltetes Potentiometer, mit dem die Ansprechempfindlichkeit geregelt werden kann. Danach gelangt das Signal über einen Kondensator zur Basis des ersten Transistors, der es verstärkt und einer Schaltstufe zuführt, deren Ausgang vom Computer abgefragt werden kann. Dieser Ausgang führt normalerweise HIGH und wird nur dann LOW, wenn er von der Signalquelle durchgesteuert wird.

Beim Aufbau ist darauf zu achten, daß der Übertrager richtig herum eingelötet

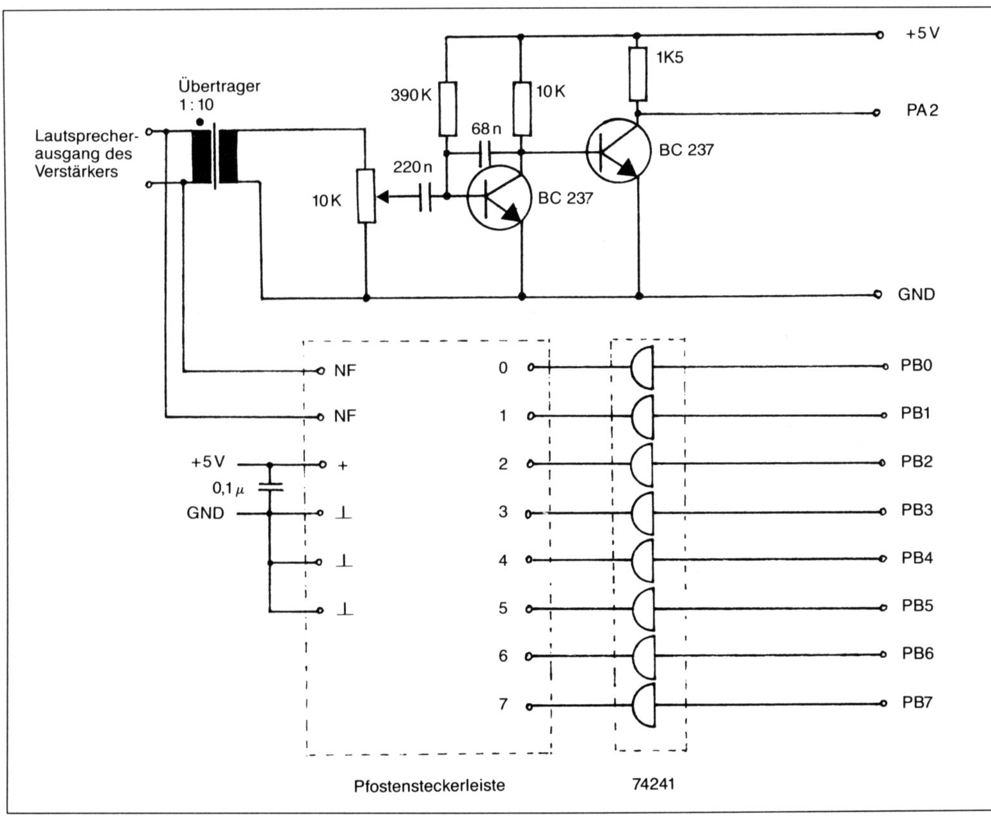


Bild 2.31: Schaltplan der Lichtorgel

wird. Er besitzt einen Farbpunkt zur Kennzeichnung der Eingangsseite (dickerer Draht).

Eine Lichtorgel ist im Prinzip ein elektronischer Schalter, der von einem Signalgeber angesteuert wird und normalerweise eine oder mehrere Glühlampen im Rhythmus der Musik aufleuchten läßt. Der zweite Kondensator in der beschriebenen Verstärkerstufe ermöglicht es, durch Gegenkopplung nur bestimmte Frequenzbereiche auszufiltern und zu verstärken. Würde man die Schaltung mehrmals mit geänderten Kondensatorwerten aufbauen, dann erhielte man getrennte Kanäle für Höhen, Mitten und Tiefen. Dabei müßten die Potis jeweils parallel zu dem Empfindlichkeitsregler des gezeichneten Kanals liegen.

Statt Glühbirnen flackern zu lassen, können farbige Effekte erzeugt werden.

2.4.8 Komparatoren

Komparatoren sind Spannungsvergleicher

1. Der zweite Eingang führt höhere Spannung als der Referenzeingang. In diesem Fall ist die Differenz positiv. Der Operationsverstärker nimmt wegen seiner großen Verstärkung schon bei der kleinsten Überschreitung der Referenzspannung seine maximale Ausgangsspannung an. Diese liegt in der Größenordnung der Betriebsspannung, also etwa +5 Volt. Damit ist der Komparatorausgang in diesem Fall HIGH.

2. Der andere Eingang führt niedrigere Spannung als der Referenzeingang. Die Differenzspannung ist nun negativ und der Operationsverstärkerausgang geht sofort auf die kleinstmögliche Spannung. Bei der angegebenen Beschaltung ist dies Massepotential mit 0 Volt, also LOW.

Bild 2.32 zeigt eine reale Beschaltungsmöglichkeit des Komparators. Das Poti (regelbarer Widerstand) dient hier als Spannungsteiler. Man kann es sich ersetzt denken durch zwei hintereinandergeschaltete Einzelwiderstände, zwischen denen

Anzeige	Gradzahl
0	0 – 22,5
1	22,5 – 45
2	45 – 67,5
3	67,5 – 90
4	90 – 112,5
5	112,5 – 135
6	135 – 157,5
7	157,5 – 180
8	180 – 202,5
9	202,5 – 225
10	225 – 247,5
11	247,5 – 270
12	270 – 292,5
13	292,5 – 315
14	315 – 337,5
15	337,5 – 360

Tabelle 2.5: Zuordnung der Anzeige zum realen Winkel

ßigkeit erkennbar. Von innen nach außen hat jeder Ring genau doppelt so viele Felder wie sein Nachbar. Dabei wird jedes Feld in zwei gleich große Bereiche mit schwarz und weiß eingeteilt. Bei acht

zwischen zwei Bereichen, dann wechseln die Zustände nicht genau gleichzeitig. Dabei können kurzzeitig Meßergebnisse auftreten, die völlig falsch sind. Dreht sich in Bild 2.35 zum Beispiel die Scheibe langsam im Uhrzeigersinn, dann sind zunächst alle vier Ausgänge HIGH. Es ergibt sich der Meßwert 15 entsprechend 337,5 bis 360 Grad. Dann werden die oberen drei Lichtschranken verdeckt. Die untere ist ja leicht verstellt und bleibt zunächst offen. Dadurch entsteht der Meßwert 8, also 180 bis 202,5 Grad; ein völlig falsches Ergebnis! Erst wieder etwas später hat sich die Scheibe auf den richtigen Wert weitergedreht: 0 für 0 bis 22,5 Grad. Besonders stark macht sich diese Unsauberkeit bemerkbar, wenn die Scheibe auf der Grenze zwischen zwei Werten stehenbleibt. Es kann in diesem Fall vorkommen, daß die Anzeige ständig zwischen mehreren – zum Teil völlig falschen – Werten hin- und herspringt.

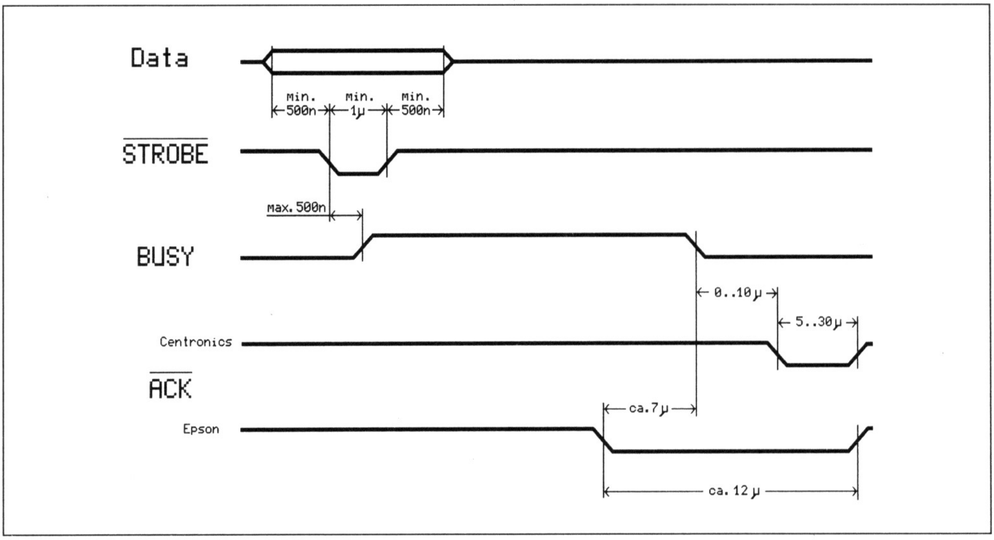


Bild 2.38: Abläufe an der Centronics-Schnittstelle bei der Übergabe eines Zeichens

buchstaben, Ziffern und Sonderzeichen. Die ersten 31 Codes bilden Steuerzeichen für Drucker und Terminals. Sie werden ebenfalls im Anhang B genauer erläutert.

2.5.2 Die Centronics-Schnittstelle

Der Druckerhersteller Centronics führte eine Schnittstelle ein, die bald von zahlreichen anderen Firmen übernommen wurde und heute einen Quasi-Standard für Drucker darstellt. Die Datenübertragung geschieht danach über acht parallele Leitungen mit Handshake-Protokoll. In Kapitel 1.2.2 wurde diese Synchronisationsart bereits kurz beschrieben. Die Meldeleitung des Computers heißt bei Centronics $\overline{\text{STROBE}}$, die des Druckers $\overline{\text{ACK}}$ (Acknowledge = Bestätigung).

Signalpin	Massepin	Signal	Richtung	Bedeutung
1	19	$\overline{\text{STROBE}}$	Eingang	Bei der fallenden Flanke von $\overline{\text{STROBE}}$ werden die Daten vom Drucker übernommen. $>1\mu s$
2	20	DATA 1	Eingang	Datenbit
3	21	DATA 2	Eingang	Datenbit
4	22	DATA 3	Eingang	Datenbit
5	23	DATA 4	Eingang	Datenbit
6	24	DATA 5	Eingang	Datenbit

Signalpin	Massepin	Signal	Richtung	Bedeutung
7	25	DATA 6	Eingang	Datenbit
8	26	DATA 7	Eingang	Datenbit
9	27	DATA 8	Eingang	Datenbit
10	28	$\overline{\text{ACKNLG}}$	Ausgang	LOW-Impuls von ca. 10μs Länge. Besagt, daß die Daten verarbeitet wurden und der Drucker wieder bereit ist.
11	29	BUSY	Ausgang	Solange BUSY HIGH-Signal führt, ist der Drucker nicht empfangsbereit.
12	30	PE	Ausgang	HIGH = Papierende
13	–	SELECTED	Ausgang	Geht nach LOW, wenn der Drucker OFF-LINE ist
14	–	$\overline{\text{AUTOFEEDXT}}$	Eingang	Linefeed-Steuerung LOW = automatisch HIGH = per Befehl
15	–	NC	–	unbenutzt
16	–	0V	–	Massepegel (Logik)
17	–	CHASSIS GND	–	Masse Drucker (isoliert von Logikmasse)
18	–	+5V	–	Versorgungsspannung
19-30	–	GND	–	Massepins für 1-12
31	–	$\overline{\text{INIT}}$	Eingang	Druckerinitialisierung bei LOW-Impuls länger 50μs
32	–	$\overline{\text{ERROR}}$	Ausgang	Geht auf LOW, wenn 1. Drucker offline, 2. Papierende, 3. Fehler entdeckt.
33	–	GND	–	Massepins wie 19-30
34	–	NC	–	unbenutzt
35	–	–	–	HIGH-Pegel
36	–	$\overline{\text{SLCTIN}}$	Eingang	Druckerselektion Codes DC1/DC3 funktionieren nur, wenn dieser Pin HIGH ist.

Tabelle 2.6: Belegung des Centronics-Steckers eines Druckers

Außer diesem Übertragungsprotokoll ist hier noch eine andere Synchronisationsmöglichkeit zwischen den Geräten vorgesehen. Verantwortlich dafür ist die Leitung BUSY, die vom Drucker bedient wird (englisch busy = beschäftigt). Die ersten beiden Schritte der Übertragung laufen genau gleich ab.

1. Der Computer legt die Kombination für das zu druckende Zeichen an die Datenleitungen und

2. gibt auf seiner Meldeleitung ($\overline{\text{STROBE}}$) einen kurzen Impuls aus, um zu signalisieren, daß die Daten anliegen.

Dann aber:

3. Der Drucker setzt zunächst die Leitung BUSY auf HIGH, liest die Kombination ein, druckt das Zeichen und

4. setzt dann erst die Leitung BUSY zurück auf LOW, um dem Computer seine Empfangsbereitschaft anzuzeigen.

Bild 2.38 gibt den genauen Zeitablauf der Übertragung eines Bytes grafisch wieder. Wie zu sehen, sind bei der Generierung des Acknowledge-Signals wiederum zwei leicht verschiedene Arten üblich. Normalerweise beginnt der $\overline{\text{ACK}}$-Impuls nachdem BUSY bereits wieder LOW wurde. Beim sogenannten Epson-Type-Handshake beginnt dieser Impuls jedoch schon vor der abfallenden Flanke von BUSY. In jedem Fall aber ist ein Zyklus erst beendet, wenn $\overline{\text{ACK}}$ wieder HIGH wird.

Tabelle 2.6 zeigt die Belegung des an Druckern verwendeten 36-poligen Steckers, dessen Pinanordnung in Bild 2.39 zu sehen ist, und gibt die Bedeutung der Signale an.

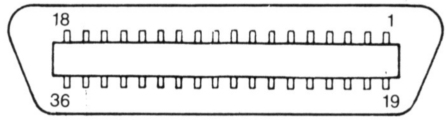


Bild 2.39: Außenansicht Centronics-Normbuchse bzw. Stecker mit Sicht auf Lötseite

Der Anschluß $\overline{\text{INIT}}$ (Signalpin 31) ist der Reseteingang des Druckers. Mit einem LOW-Impuls kann er hier von Hand auf die vorgewählten Werte zurückgesetzt werden. Sie können dazu die Tasterschaltung nach Bild 2.20 verwenden, die unter Umständen sogar noch ins Steckergehäuse paßt.

Pin 18 führt bei manchen Druckern (zum Beispiel STAR) die +5 Volt-Versorgungsspannung der internen Elektronik. Das ist sehr nützlich für kleine Interface-Schaltungen, die so – ohne zusätzliche Stromversorgung – einfach in die Zuleitung gebaut werden können.

Es ist nicht erforderlich, alle Schnittstellensignale zu erzeugen bzw. auszuwerten. Für einen einwandfreien Betrieb genügen bereits die acht Datenbits sowie die Signale $\overline{\text{ACK}}$ bzw. BUSY und $\overline{\text{STROBE}}$. Alle anderen Pins können unbeschaltet bleiben. Sie sind bei vielen Druckern teilweise leider auch von Hersteller zu Hersteller unterschiedlich belegt.

2.5.3 Ein universelles Druckerkabel

Der Anschluß eines Centronics-Druckers an den Amiga ist einfach:

Man nehme einen 25-poligen Min-D-Stecker auf der einen Seite (für Amiga 1000 weiblich, sonst männlich), den beschriebenen 36-poligen Amphenol-Stecker auf der anderen, dazwischen ein mindestens 11-poliges Kabel von höchstens 1,5m Länge, am besten mit Abschirmgeflecht.

Tabelle 2.7 gibt die Verbindungen für ein universelles Druckerkabel an, das sich in der Praxis mit Druckern verschiedenster Hersteller optimal bewährt hat. Dabei wird die BUSY-Leitung des Druckers nicht verbunden. Die Schnittstellen-Software im Amiga fragt sie ohnehin nicht ab und benutzt allein das Signal $\overline{\text{ACK}}$.

ist sichergestellt, daß an ihr kein Potentialabfall entlang des Kabels auftreten kann. Ebenso sollten eventuell überzählige Adern an diesem Ende mit an die Masse gelegt werden, während sie auf der anderen Seite unbeschaltet bleiben. Diese Maßnahme verstärkt den Abschirmeffekt.

Doch bevor Sie sich an die Arbeit machen, lesen Sie lieber noch ein Stückchen weiter. Der Teufel steckt nämlich im Detail und der Parallelport Ihres Amiga steht auf dem Spiel!

2.5.4 Stromschnellen

Nehmen Sie die Begrenzung der Kabel-

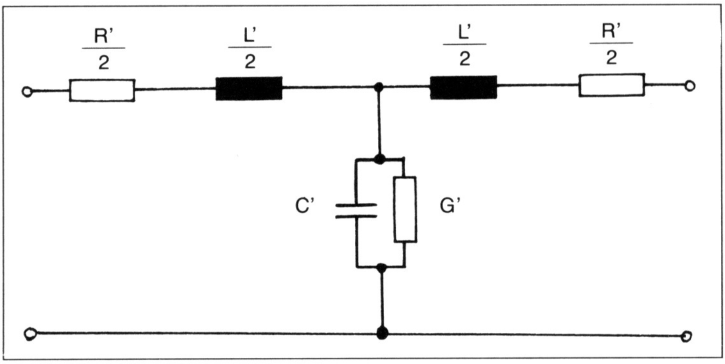


Bild 2.40: Ersatzschaltung eines Leitungselements

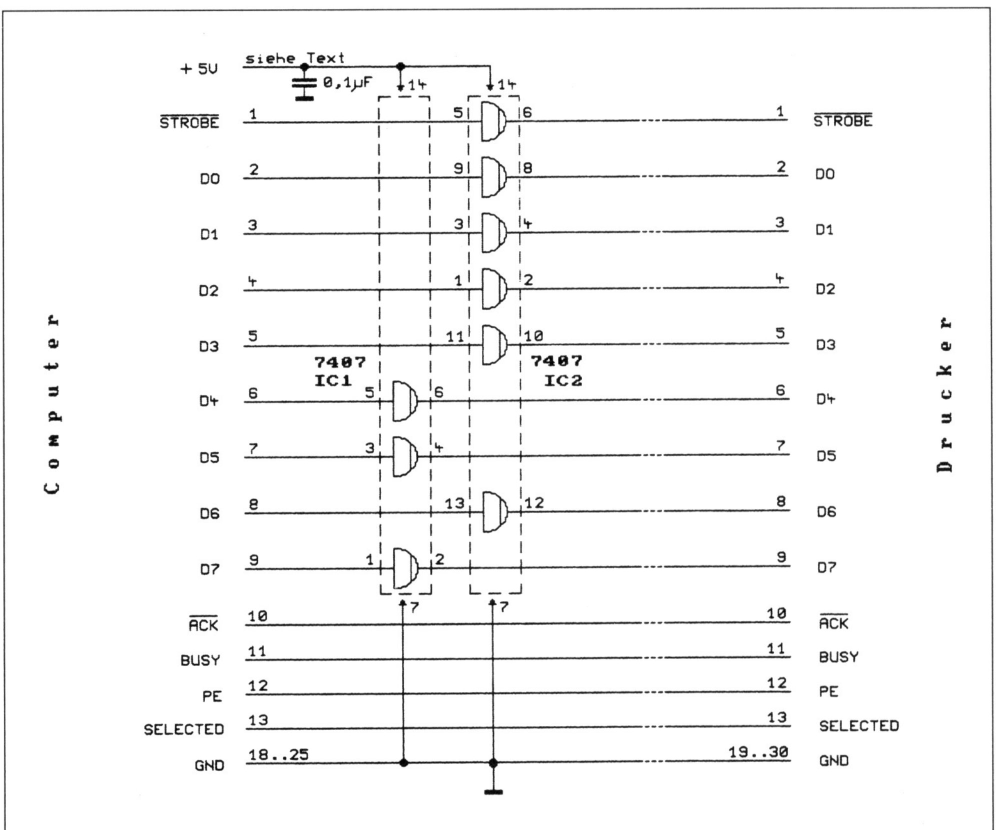


Bild 2.41: Die schaltung des Druckertreibers

dicht nebeneinander geführt, nur durch ihre Isolierungen getrennt. Sie bilden damit einen Kondensator. Jetzt wechselt die Datenleitung innerhalb sehr kurzer Zeit von 0 Volt auf +5 Volt. Der Kondensator muß sich sehr schnell aufladen und entnimmt dabei dem treibenden Port im ersten Moment einen sehr großen Strom. Genau das gleiche passiert auch wieder beim Umladen in den Ausgangszustand. Je größer die Kabellänge und damit der Wert des gebildeten Kondensators ist, um so größer wird auch die Strombelastung für den Ausgangstreiber. Irgendwann einmal brennt er durch, und dann ist guter Rat im wahrsten Sinne des Wortes teuer.

Sie brauchen aber nun nicht gleich zu verzweifeln oder das lange Kabel abzuschneiden und den Drucker näher heranzurücken, denn mit externen Treiberbausteinen läßt sich ja dem I/O-Baustein etwas Arbeit abnehmen. Besonders geeignet sind nicht-invertierende Open-Collector-Treiber vom Typ 7407 oder 7417. Bild 2.41 gibt eine geeignete Schaltung an. Für jede Ader sind damit am Ausgang Ströme von bis zu 40mA zulässig. Steuert man diesen Treiber über ein kurzes Kabel direkt mit dem Amiga an, dann darf der Drucker auch ruhig gute fünf Meter weiter stehen. Und sollte wirklich einmal etwas schief gehen, dann sind allerhöchstens die beiden Treiberbausteine hin, und Sie mit fünf Mark wieder dabei.

Die bei Open-Collector-Schaltungen zur Funktion nötigen Pull-up-Widerstände sind standardmäßig bereits im Drucker enthalten, da es bei langen Übertragungsleitungen vorteilhaft ist, diese Widerstände erst auf der Empfängerseite einzufügen.

2.5.5 Der kleine Unterschied

Die Herstellung des Kabels mit den eingeschleiften Treibern ist sehr einfach, wenn zuerst die kleine Platine nach Bild 2.42 angefertigt wird. Bild 2.43 zeigt den Bestückungsplan. Die wenigen Bauteile sind in Tabelle 2.8 zusammengestellt. Allerdings gibt es Unterschiede bei den Steckverbindungen. Für den Amiga 1000 muß ein weiblicher 25-poliger Sub-D-Stecker verwendet werden. Er besitzt eine vom IBM-Standard abweichende Belegung. Trotzdem ist die gleiche Platine einsetzbar. Einzig und allein der Sub-D-Stecker muß hier von der Leiterseite her eingelötet werden. Das mag auf den ersten Blick etwas unschön erscheinen, ist aber durchaus praktikabel. Von den vielen Massepunkten an der je nach Steckerausführung eventuell schwer zugänglichen 12-poligen Seite genügt die Verlötung eines einzigen. Bei allen anderen Modellen wird ein entsprechender männlicher Stecker wie üblich von der Bauteilseite her eingesetzt. Vergessen Sie bei der Bestückung nicht die eingezeichnete Brücke.

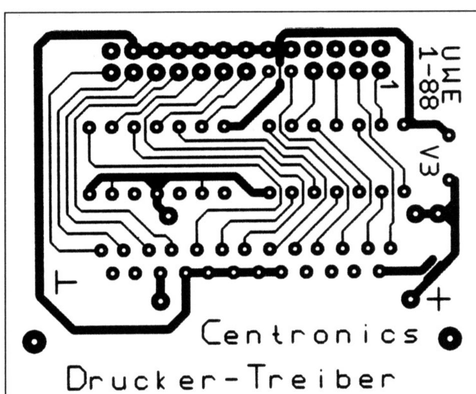


Bild 2.42: Layout für den Centronics-Druckertreiber

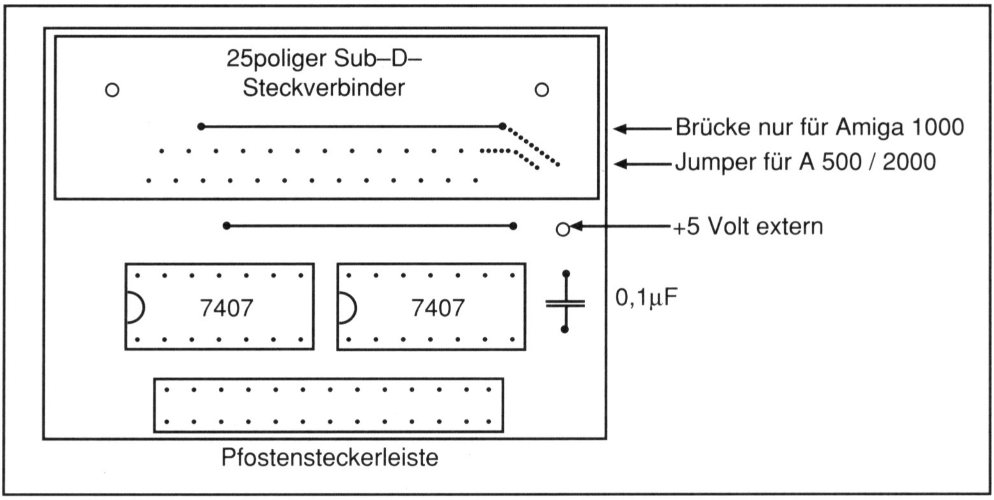


Bild 2.43: Bestückung des Centronics-Druckertreibers

Eine gewisse Schwierigkeit stellt nun noch die Spannungsversorgung der Zusatzplatine dar. Die Amiga-Modelle 500 und 2000 liefern an Pin 14 der erwähnten 25-poligen Steckverbindung +5 Volt, allerdings über einen 47-Ohm- Widerstand. Soll die Platine an einem dieser Geräte betrieben und aus dem Amiga mit Strom versorgt werden (und wirklich nur dann!), ist der schrägliegende Jumper auf der Treiberplatine mit einer Lötzinn-Brücke zu schließen. Dazu ist allerdings im Gerät der Sicherheitswiderstand zu überbrücken. Lesen Sie zuerst Kapitel 2.1.2. Falls Sie das nicht möchten, kann über den entsprechend gekennzeichneten Punkt an der Platine +5 Volt von außen eingespeist werden.

Der Amiga 1000 liefert direkt +5 Volt, allerdings an Pin 23 des Parallelports. Nur in diesem Fall ist auf der Platine die entsprechend gekennzeichnete Brücke einzulöten.

Nun folgt das eigentliche Kabel. Normalerweise fände dazu ein abgeschirmtes Rundkabel Verwendung. Das mühsame und zeitaufwendige Abisolieren, Verzinnen und Verlöten der vielen Adern ist aber nicht jedermanns Sache. Daher wurde hier ein Flachbandkabel mit Quetschverbindungen vorgesehen. Auf der Platine befindet sich eine 26-polige Pfostensteckerleiste. Ihr Gegenstück wird an das eine Ende des Flachbandkabels montiert, der Centronics-Stecker kommt an das andere Ende. Hier werden nur die Pins 1 bis 13 und 19 bis 31 benutzt. Achten Sie darauf, daß beide Stecker von der gleichen Seite her angepreßt werden. Jede zweite Ader des Flachbandkabels liegt an Masse. Das ergibt eine genügend große Abschirmwirkung. Einige Masse-Pins bleiben auf der Treiberplatine unbeschaltet, weil bei manchen Druckern an den entsprechenden Anschlüssen andere Signale anliegen.

1	Sub-D-Stecker, 25-pol, abgewinkelte Kontakte (siehe Text)
2	Open-Collector-Treiberbausteine 7407 bzw. 7417
2	IC-Sockel, 14-pol
1	Kondensator 0,1 Mikrofarad, Keramik
1	Pfostensteckerleiste, 26-pol, doppelreihig
1	Pfostenfederleiste, 26-pol, anpreßbar
1	Flachbandkabel, 26-pol
1	Centronics-Stecker, 36-pol, anpreßbar
1	einseitige Platine nach Bild 2.42

Tabelle 2.8: Der Einkaufszettel zum Druckertreiber

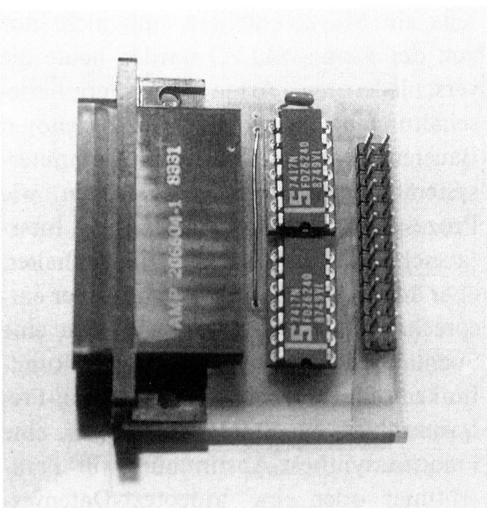


Foto 2.1: Druckertreiber

Eine andere Variante dieses Druckertreibers wird auch die nächste Platine quasi ganz nebenbei enthalten. Doch dazu soll zunächst einmal weit ausgeholt werden.

2.6 I²C-Bus am Amiga

Neue Bausteine an ein Computersystem anzuschließen ist oft schwierig und für Laien kaum zu schaffen. Hier soll ein System vorgestellt werden, mit dem sich die Vorteile spezieller ICs voll nutzen lassen, obwohl sie nur mit ein paar Drähtchen am Amiga angeschlossen werden.

2.6.1 Computer-Verbindungswege

Der Ausdruck »Bus« bezeichnet in der Computertechnik grundsätzlich ein Leitungssystem zum Austausch von Informationen zwischen mehreren Sendern und Empfängern. Es gibt viele unterschiedliche Realisierungsmöglichkeiten für diese Verkehrswege. Jede hat dabei Vor- und Nachteile. Bekannte genormte Bussysteme sind etwa der IEC-Bus oder der in professionellen Steuersystemen oft eingesetzte parallele VME-Bus. Bei einer parallelen Lösung werden viele Daten gleichzeitig übertragen, während sie bei einem seriellen Konzept zuerst aufgeteilt werden, und dann nacheinander über das Kabel gehen müssen. Eine parallele Übertragung kann wesentlich schneller sein. Dem steht allerdings ein deutlich geringerer Hardwareaufwand bei der seriellen Übermittlung gegenüber. Nicht nur das Kabel allein ist hier maßgebend (viele Adern können bei

langen Strecken schon recht teuer sein), sondern vor allem die Geräte selbst kommen mit weniger Aufwand aus. Außer den üblichen Steueradern wird nur noch eine einzige Datenleitung benötigt. Das wirkt sich auf die Pinzahl und damit auf die Gehäusegröße der eingesetzten ICs aus. Bei kleineren Komponenten läßt sich auch die benötigte Platinenfläche der Schaltung verkleinern. Das ist besonders wichtig, da sie einen beträchtlichen Anteil an den Gesamtkosten eines Systems ausmacht.

Die Stromaufnahme der Schaltung sinkt, unter Umständen kann das Netzteil kleiner werden und eventuell sogar das ganze Gerät. Auf jeden Fall wird die Herstellung billiger.

Außerdem hat die Erfahrung gezeigt, daß die Störanfälligkeit eines Systems mit der Anzahl der Steckverbindungen wächst.

Aus allen diesen Erkenntnissen wurden zuerst in der Unterhaltungselektronik Konsequenzen gezogen, einem Gebiet, auf dem man zunächst den intensiven Einsatz von Mikrocomputersystemen

tronik setzt. Gerade in diesem mächtigen Industriezweig aber, der die schwere Aufgabe hat, unter starkem Konkurrenzdruck möglichst preisgünstig hochtechnisierte Produkte herzustellen, die immer höheren Ansprüchen genügen müssen, hat sich in den letzten Jahren ein deutlicher Trend zur digitalen Datenverarbeitung bemerkbar gemacht. Kein Wunder also, daß der neue I^2C-Bus bald in vielen Geräten Verbreitung fand.

Inzwischen hat sich dieses Bussystem bereits am Markt etabliert, und nicht nur von der Firma VALVO werden heute die verschiedensten Arten von Peripherieschaltungen angeboten. Dazu gehören Bausteine, wie sie in normalen Computersystemen auch Verwendung finden, wie Prozessoren, Speicherbausteine, Interfaceschaltungen und Leistungsschalter, aber auch ganz spezielle ICs mit einer entsprechenden Schnittstelle, wie etwa eine Suchlauf-Interfaceschaltung für Rundfunkempfänger mit AM/FM-ZF-Frequenzzähler, ein Stereo-Klangregler, eine Frequenzsynthese-Abstimmung für Fern-

Auch im Audiobereich hält der Computer immer stärker Einzug. Man denke nur an die Realisierung der komplexen Steuerungsmechanismen eines CD-Spielers. Neuerdings werden Bausteine mit I^2C-Schnittstelle sogar in PKWs eingebaut. Der Zwang, noch leistungsfähigere, aber doch kompakte und vor allem preiswerte und flexible Produkte zu fertigen, drängt die Entwickler immer stärker zum Einsatz von solchen miniaturisierten Mikrocomputersystemen. Bei den geringen benötigten Datenmengen pro Zeit in den angesprochenen Einsatzgebieten stellt dabei die begrenzte Übertragungsgeschwindigkeit noch nicht einmal unbedingt einen Nachteil dar.

Bild 2.44 gibt einen Einblick in die vielseitige Verwendbarkeit des I^2C-Busses bei herkömmlichen Analogsystemen. Als Beispiel wird die Steuerung eines Komplettsystems gezeigt, bestehend aus Fernseh- und Rundfunkempfänger mit Video- und Bildschirmtextteil, Anzeigen und Uhr.

Der im folgenden beschriebene Schaltungsteil ermöglicht es, Bausteine mit I^2C-Schnittstelle auch am Amiga zu betreiben. Das eröffnet viele Chancen, einerseits das breite Angebot an interessanten Bausteinen mit I^2C-Schnittstelle selbst zu nutzen, und andererseits mit dem Heimcomputer einmal in ein kommerzielles Gerät hineinzuschnuppern, und eventuell sogar dessen Steuerung von einem selbstgeschriebenen Programm übernehmen zu lassen!

Als Beispiel soll hier die Ansteuerung einer Uhren-/Kalenderschaltung dienen, mit der ständig, auch wenn der Computer zwischendurch mal abgeschaltet wurde, die aktuelle Uhrzeit zur Verfügung steht.

2.6.2 Busverkehr

Der I^2C-Bus ist für den bidirektionalen 2-Draht-Datenverkehr zwischen verschiedenen integrierten Schaltungen oder Modulen ausgelegt. Die beiden Leitungen sind eine serielle Datenleitung (SDA) und eine serielle Taktleitung (SCL). Bild 2.45 zeigt die prinzipielle Zusammenschaltung von verschiedenen Teilnehmern. Ein Baustein, der am I^2C-Bus eine Information erzeugt und aussendet, heißt »Sender«; eine Einheit, die eine Information entgegennimmt, ist ein »Empfänger«. Die Schaltung, die eine Übertragung von Informationen steuert, wird als »Master« bezeichnet, die vom Master gesteuerten Schaltungen heißen »Slaves«. Der Master generiert ein Taktsignal auf der seriellen Clock-Leitung SCL. Während eines jeden

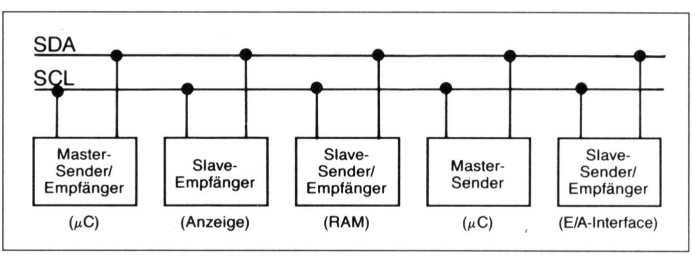


Bild 2.45: Verschiedene Bausteine am I^2C-Bus

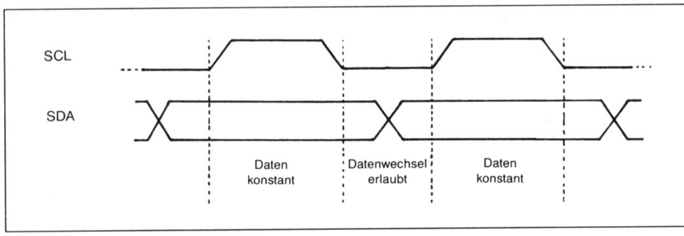


Bild 2.46: Abhängigkeit der Datenleitung SDA von der Taktleitung SCL

Taktimpulses wird ein Datenbit übertragen. Bild 2.46 macht das deutlich.

Ein Datenwechsel darf nur während der LOW-Phase des Taktsignals erfolgen. Die einzige Abweichung von dieser Regel bildet der Anfang einer Übertragung. Ein Wechsel der Datenleitung von HIGH nach LOW während die Taktleitung HIGH-Potential führt, gilt als Startbedingung und leitet jede Busübertragung ein.

Es können durchaus mehrere Master in einem System existieren, es darf aber nur immer einer davon den Bus belegen. Ein Slave-Baustein kann Sender sein, wenn er von einem Master angesprochen wurde und Antwort gibt. Nach wie vor kommt jedoch das Taktsignal dann vom jeweiligen Master.

Ermöglicht wird die Mehrfachnutzung der Leitungen durch Open-Collector-Schaltungstechnik. Dabei existiert für alle Bausteine pro Busleitung ein gemeinsamer Pull-up-Widerstand. Jeder Bus-Sender kann über einen Ausgangstransistor die Leitung nach LOW ziehen. Bei nicht belegtem Bus verbleiben sowohl die Daten- als auch die Taktleitung im HIGH-Zustand.

Die Anzahl der bei einem Buszugriff übertragbaren Bytes ist nicht begrenzt. Jedes Byte besteht aus acht Bit, denen ein Quittierbit (Acknowledge) folgt. Bild 2.47 zeigt die genauen Abläufe bei der Übertragung des ersten Bytes. Zunächst erzeugt der Sender eine Startbedingung. Die Taktleitung geht daraufhin auf LOW und die Spannung für das erste Bit kann auf

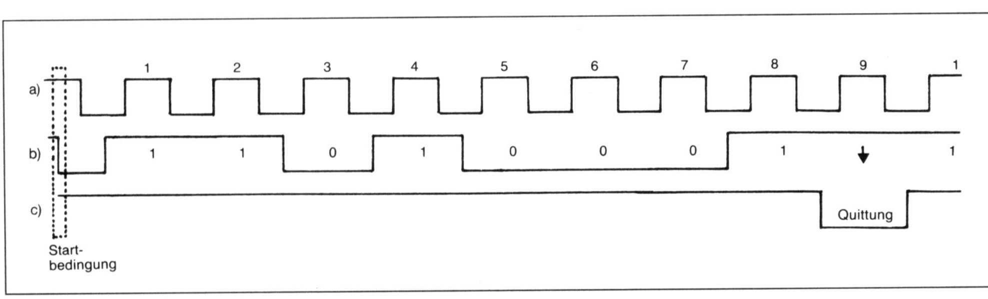
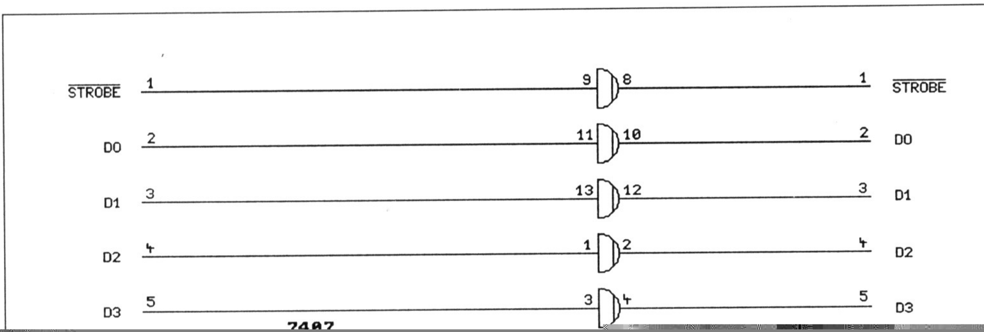


Bild 2.47: Beispiel einer Datenübertragung (Adreßbyte lesen). a) Taktsignal SCL b) Daten SDA vom Sender (Computer) c) Datenleitung vom Empfänger (PCB 8573)


STROBE
D0
D1
D2
D3
7407

Transistor zieht nun BUSY bei HIGH-Pegel von SDA nach LOW, so daß die Datensignale dort invertiert erscheinen. Die so verschaltete BUSY-Leitung darf nicht zum Drucker durchverbunden werden.

Viele I²C-Bausteine besitzen einen Interrupt- oder Meldeausgang mit offenem Kollektor. Damit er auch benutzt werden kann, wurde eine Signalisierungsmöglichkeit über $\overline{\mathrm{ACK}}$ vorgesehen. Bei diesem Signal handelt es sich um einen Open-Collector-Ausgang des Druckers, der normalerweise HIGH ist, und mit einem kurzen LOW-Impuls anzeigt, daß die anliegenden Daten verarbeitet wurden. Die Leitung kann also zusätzlich auch von anderen Open-Collector-Quellen bedient werden. Eine negative Flanke am Anschluß $\overline{\mathrm{ACK}}$ setzt im Amiga das FLAG-Bit des zuständigen 8520. Damit kann bei geeigneter Programmierung auch ein Interrupt ausgelöst werden.

2.6.4 I²C-Software

Die genannten Abläufe werden alle von einer universellen Handlingroutine erledigt, von der aus sich die Bus-Operationen leicht und übersichtlich steuern lassen. Sie macht den Amiga mit der I²C-Schnittstelle zu einem Master-Sender/Empfänger. Listing 2.1 zeigt den Hauptteil des C-Quellprogramms I²C.c. Der Routine wird beim Aufruf ein Zeiger auf den Bereich mitgegeben, in dem die zu sendenden Daten stehen sowie deren Anzahl und die Anzahl der zu empfangenen Antwortbytes. Sie liefert einen Zeiger auf den Bereich zurück, in dem die Antwort steht.

```
char *i2c_rw ( str, sc, rc )  /* this function does read-write to i2c-bus */
     char *str;               /* pointer to the string to be sent        */
     int sc,rc;               /* sc : number of bytes to be sent         */
                              /* rc : number of bytes to be received     */
                              /* returns a pointer to the received string */
{
int i,iw,j,errcn,err=0;
char c;
static char res[32];

RESC;                         /* reset clock and set data */
SETD;
SETC;
delay();                     /* wait for signals to be stabil */

RESD;                         /* send startcondition */
for (i=0;i<sc;i++)
   {
   c=*(str+i);
   for (j=0;j<8;j++)          /* now send one byte */
      {
```

```
      RESC;
      if ((c>>(7-j))&1) SETD
      else RESD;
      SETC;
      };
   RESC;
   SETD;
   SETC;                        /* wait for acknowledge */
   for (errcn=0;(errcn<10)&&(DATA=='1');errcn++)for(j=0;j<10;j++);
   if (errcn==10)
      {                         /* got no acknowledge, so stop transmission */
      err=1;
      RESC;
      goto stop;
      };
   RESC;
   };                           /* all bytes sent */
if (rc>0)                       /* is there any byte to be received */
   {
   for (i=0;i<rc;i++)
      {
      for (j=0;j<8;j++)         /* read one byte */
         {
         SETC;
         c=(c<<1)(DATA&1);
         RESC;
         };
      res[i]=c;
      RESD;
      if (i+1<rc)
         {
         SETC;                  /* generate acknowledge if there are anymore */
         RESC;                  /* bytes to be received                      */
         SETD;
         };
      };
   SETD;
   SETC;
   RESC;
   };
stop:;
RESD;                           /* send stopcondition */
SETC;
```

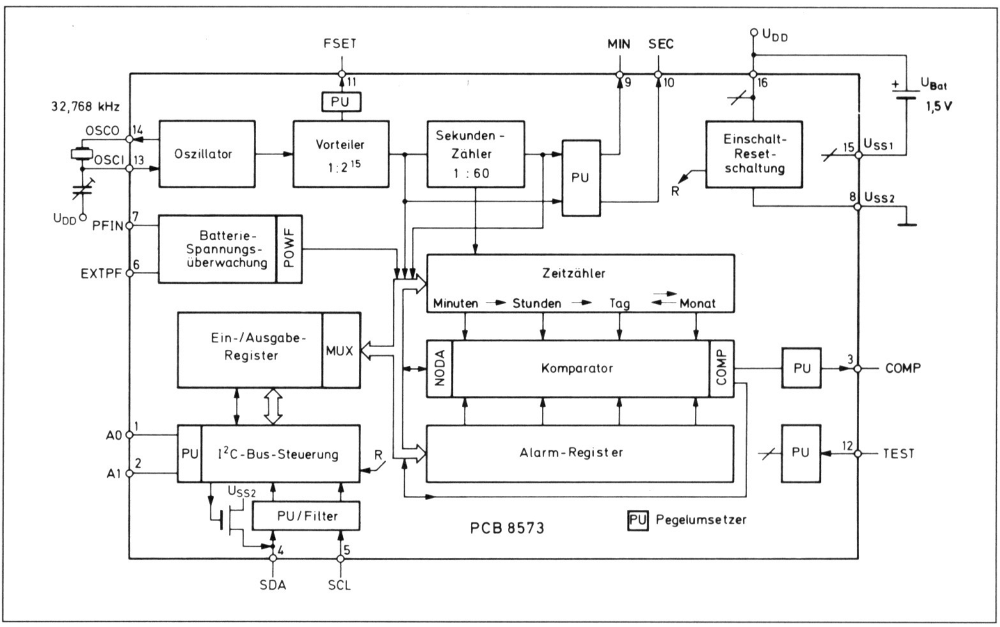
```
SETD;
if (err!=0)
   {
   panic("no acknowledge\n");
   return(0L);                /* return NULL on failure */
   };
return (res);
}
```

Listing 2.1: Unterroutine zur Bedienung des I²C-Busses

Das Programm I²C gibt Ihnen die Möglichkeit, am Bus angeschlossene Bausteine einfach anzusprechen. Der Aufruf ist:

`I²C ANTWORTBYTES BYTE1 BYTE2 BYTE3 ...`

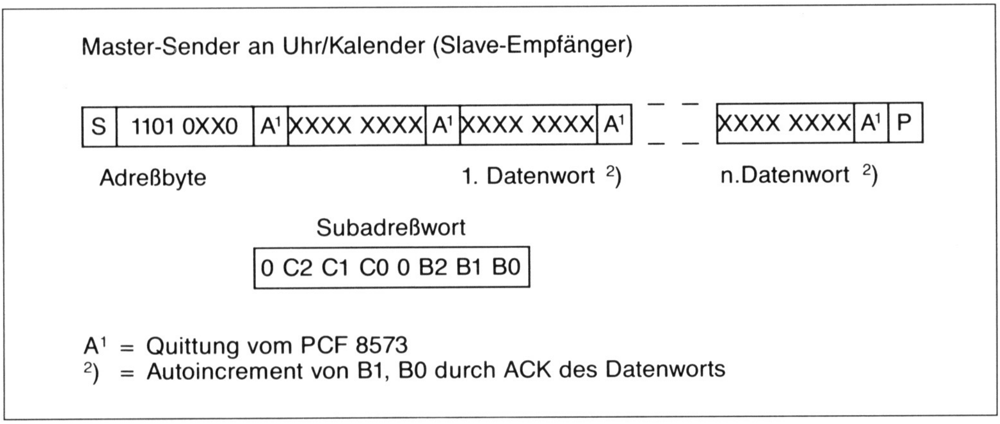
ANTWORTBYTES ist die Anzahl der Datenbytes, die Sie empfangen wollen. Danach folgen die zu sendenden Bytes. Die Verwendung des Programms wird gleich anhand eines Beispiels erläutert.



Adresse des PCB 8573
1. Byte nach der Startbedingung:

MSB							LSB
1	1	0	1	0	A1	A0	R/$\overline{W}$

Bild 2.49: Innerer Aufbau und Adreßwort der Uhren-Kalender-Schaltung



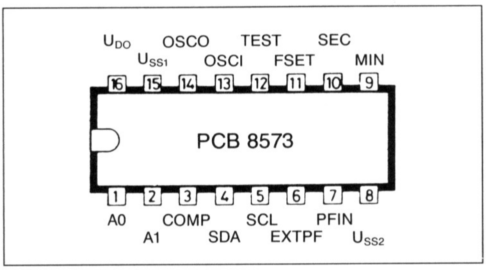


Bild 2.50: Pinbelegung des Schaltkreises

2.6.5 Auf der Höhe der Zeit

Unter den Bausteinen mit I^2C-Schnittstelle befindet sich eine Echtzeituhr mit eingebautem Kalender, die mit sehr wenig Strom bei kleiner Spannung auskommt, und die intern auch weiterläuft, wenn der Rechner ausgeschaltet ist. Es handelt sich um den Baustein PCF 8573, dessen groben Aufbau Bild 2.49 zeigt. Bild 2.50 enthält seine Pinbelegung.

Die integrierte Schaltung verfügt über Zeitzähler und Alarmregister, jeweils für Minute, Stunde, Tag und Monat. Die Uhr selbst wird von einem Akku versorgt. Aufgrund der verwendeten CMOS-Technologie beträgt die maximale Stromaufnahme im Ruhebetrieb nur etwa 10 μAmpere, das heißt, die Uhr würde auch noch nach mehr als 5 Jahren laufen, ohne daß der Rechner ein einziges Mal eingeschaltet werden müßte, um den Akku nachzuladen. Die 5V-Speisespannung wird nur für den Datenverkehr und die Ausgabe von Impulsen benötigt. Dazu dienen auch die Pegelumsetzer PU.

Als Zeitbasis wird ein Quarzoszillator benutzt, dessen Frequenz von 32,768kHz im Vorteiler durch 2^{15} geteilt wird. Die daraus resultierenden Sekundenimpulse steuern den Zeitzähler an. Dort werden sie von einem – nicht lesbaren – Sekundenzähler summiert und nach jeweils 60 Sekunden an die weiteren Stufen für Minute, Stunde, Tag und Monat übertragen. Wie aus Tabelle 2.9 zu ersehen ist, umfaßt der Zählzyklus 24 Stunden. Im

Zähler	Zählerzyklus	Übertrag bei Zählerstand	Stand des Manatszählers
Minuten	00...59	59 → 00	1 ... 12
Stunden	00...23	23 → 00	1 ... 12
Tage	01...28	28 → 01	2
	oder 01...29	29 → 01	2
	01...30	30 → 01	4,6,9,11
	01...31	31 → 01	1,3,5,7,8,10,12
Monate	01...12	12 → 01	

Tabelle 2.9: Zählzyklen und Überträge des PCB 8573

Zeitzähler wird automatisch, je nach Monatslänge, ein Zyklus von 28, 30 oder 31 Tagen berücksichtigt.

Die genannten Zähler können über den I^2C-Bus gesetzt und gelesen werden. An den Ausgängen SEC und MIN werden Sekunden- bzw. Minutenimpulse ausgegeben, zum Beispiel für die Steuerung eines Blinkindikators im Sekundentakt. Die Zustände dieser Anschlüsse können auch über den Bus gelesen werden. In das über den I^2C-Bus setz- und lesbare Alarmregister kann man eine Schaltzeit speichern, die mit der aktuellen Zeit ständig verglichen wird. Tritt Gleichheit auf, dann setzt der Baustein eine Marke, die als Steuersignal am Bus, wie auch am Anschluß COMP zur Verfügung steht, bis sie mit einem speziellen Befehl gelöscht wird. In Abhängigkeit von einer weiteren Marke NODA, die ebenfalls über den Bus beeinflußt werden kann, erfolgt dieser Vergleich wahlweise mit oder ohne Berücksichtigung des Datums.

Die letzte Marke POWF (Powerfail) dient zum Aufdecken von Unzuverlässigkeiten in der Akku-Spannungsversorgung. Hat diese einen bestimmten Grenzwert unterschritten, so wird POWF gesetzt. Erst ein Schreibbefehl in die Zeit- oder Alarmregister löscht die Marke wieder.

Die äußere Beschaltung des Uhren-ICs ist relativ einfach. Mit der Transistorstufe wird die beschriebene Spannungskontrolle realisiert. Die im Schaltplan eingezeichnete Diode verhindert einen Stromfluß vom Akku durch den Rechner und damit eine unnötig schnelle Entladung im Ruhebetrieb.

Wie bereits erwähnt, muß mit dem ersten Byte einer Datenübertragung beim I^2C-Bus immer die angesprochene Schaltung adressiert werden.

Bild 2.49 zeigt das Adreßbyte der Uhren-/Kalenderschaltung PCF 8573. A0 und A1 werden durch zwei Hardware-Anschlüsse bestimmt, die auf der vorgestellten Platine gemeinsam auf Masse liegen. Hier muß also in beiden Fällen eine 0 stehen.

Zu erkennen ist, daß das niederwertigste Bit des Adreßwortes ($R/\overline{W}$) Einfluß auf die Betriebsart hat. Ist es 0, arbeitet die Schaltung nach der Übertragung des Adreßbytes weiterhin als Empfänger und nimmt das folgende Byte, das sogenannte Subadreßwort, entgegen. Ist $R/\overline{W}$ jedoch 1, schaltet der Baustein beim zweiten Byte

auf Senden um und gibt selbst Informationen aus.

Mit dem Subadreßwort kann gezielt ausgewählt werden, welches Register der Schaltung PCF 8573 im folgenden gelesen oder beschrieben werden soll. Mögliche Quellen bzw. Ziele sind etwa Zeitzähler »Minuten«, Alarmregister »Tage« usw.

In der Subadresse können auch Steuerbefehle übertragen werden. Dazu ist das Subadreßwort in vier Steuerbits (C3...C0) und vier Adreßbits (B3...B0) aufgeteilt. Bild 2.50 macht das deutlich. Die Subadreßworte für die einzelnen Funktionen sind in Tabelle 2.10 zusammengefaßt.

2.6.6 Beispiele für Lesen und Stellen der Uhr

Wenn Sie aus dem Uhren-/Kalenderbaustein Informationen lesen wollen, müssen Sie zunächst genau bestimmen, mit welcher Information zu beginnen ist. Nehmen wir einmal an, uns interessieren die Stunden und Minuten der laufenden Zeit. Dazu müssen wir mittels der Subadresse den Adreßzeiger auf das Stundenregister stellen.

Zuerst einmal wird das Adreßbyte des PCF 8573 für Schreiben gesendet:

`11010000`, also hexadezimal `D0`.

zu Takt	1	2	3	4	5	6	7	8	Tabelle 2
	C3	C2	C1	C0	B3	B2	B1	B0	Funktion
	0	0	0	0	0	0	0	0	Lade Adresse für Zeitzähler-Stunde
	0	0	0	0	0	0	0	1	-Minute
	0	0	0	0	0	0	1	0	-Tag
	0	0	0	0	0	0	1	1	-Monat
	0	0	0	0	0	1	0	0	Lade Adresse für Alarmregister-Stunde
	0	0	0	0	0	1	0	1	-Minute
	0	0	0	0	0	1	1	0	-Tag
	0	0	0	0	0	1	1	1	-Monat
	0	0	0	1	X	X	X	X	Lese-Zustand MIN, SEC, NODA, COMP, POWF
	0	0	1	0	X	X	X	X	Verteiler- und Sekundenzähler-Reset (ohne Minutenauf-/-abrundung)
	0	0	1	1	X	X	X	X	Sekundenzähler-Reset [1]) (mit Minutenauf-/-abrundung)
	0	1	0	0	X	X	X	X	Lösche NODA [2])
	0	1	0	1	X	X	X	X	Setze NODA [2])
	0	1	1	0	X	X	X	X	Lösche COMP [2])

[1]) Bewirkt eine Zeitkorrektur um maximal ± 30 s. War der Sekundenzähler <30, so erfährt der Minutenzähler keinen Übertrag. Ein Sekundenzähler >30 ergibt einen Übertrag in den Minutenzähler (Aufrundung)

[2]) siehe »Komparator«

Tabelle 2.10: Bedeutung des Subadreßworts beim PCB 8573

Danach folgt das Subadreßbyte für Zeitzähler-Stunde. Dies ermitteln Sie aus Tabelle 2.10 ganz oben:

`00000000`, also hexadezimal `00`.

Die beiden gefundenen Werte werden durch das Kommando

`I²C 0 D0 00`

übermittelt. Der Uhren-/Kalenderbaustein sendet keine Werte zurück, aber nun steht sein interner Adreßzeiger auf dem gewünschten Wert.

Daraufhin fangen wir neu an und senden die Adresse des PCF 8573 zum Lesen:

`11010001`, also hexadezimal `D1`.

Fordern Sie den Uhren-/Kalenderbaustein beim sich daraus ergebenden Aufruf

`I²C 8 D1`

ruhig einmal auf, mit acht Datenbyte zu antworten. Das Format, in dem die gelesenen Ziffern kodiert sind, zeigt Tabelle 2.11.

Es handelt sich hier um BCD-Zahlen. Das ist die Abkürzung für Binary Coded Decimals, also binär kodierte Dezimalzahlen. In den oberen und unteren vier Bits des Datenwortes, den beiden Nibbles (Halbbytes), werden dabei jeweils die Zehner und Einer einer zweistelligen Zahl gespeichert. Im ersten Byte erhält man so die ausgelesenen Stunden. Die Dezimalzahl 10 beispielsweise, wird nun nicht mehr dargestellt durch die achtstellige Dualzahl

`00001010` ,

sondern durch die Darstellung der 1 in der oberen Hälfte des Datenworts, und einer 0 in der unteren Hälfte, also:

```
 1    0

0001 0000 .
```

Die hexadezimale Darstellungsweise zeigt also direkt den richtigen Wert. Doch sehen Sie sich das nächste empfangene Byte an. Hier tritt bereits eine angenehme Eigenschaft des PCF 8573 zutage: Werden mehrere Bytes ausgelesen, so erhöhen sich die beiden Adreßbytes B0 und B1 (siehe Tabelle 2.10) selbsttätig durch den Acknowledge-Impuls des Empfängers. So kommt es, daß ohne eine zusätzliche Anweisung der Inhalt des Minuten-Registers folgt. Ebenso verhält es sich mit den Tagen und Monaten, doch anschließend wird nicht der Zustand des Adreßbits B2 verändert, sondern es erscheint wieder der Inhalt des Stunden-Registers und so fort, so lange, bis kein Acknowledge mehr den Registerzeiger erhöht und vom Master –

zu Takt / Datenwort aus	Zehner 1 UD	2 UC	3 UB	4 UA	Einer 5 LD	6 LC	7 LB	8 LA
Stunde	0	0	d	d	d	d	d	d
Minute	0	d	d	d	d	d	d	d
Tag	0	0	d	d	d	d	d	d
Monat	0	0	0	d	d	d	d	d
Zustand von	0	0	0	MIN	SEC	NODA	COMP	POWF

d ≙ Datenbits

Tabelle 2.11: Kodierung der Informationen beim PCB 8573

a) Lesevorgang mit Setzen der Subadresse:

S	1101 0XX0	A¹	0000 0000	A¹	S	1101 0XX1	A¹	00XX XXXX	A	0XXX XXXX	$\overline{A}$	P

Adreßwort Schreiben — Subadreßwort »Zeitzähler-Stunde« — Startwiederholung — Adreßwort Lesen — Datenwort »Stunde« ²) — Datenwort »Minute« ²)

A¹ = Acknowledge
$\overline{A}$ = kein Acknowledge vom Empfänger
²) = automatische Erhöhung der Adresse

I. Es gibt zwei verschiedene Lesevorgänge (Protokolle).
b) Lesevorgang ohne Setzen der Subadresse: ³)

S	1101 0XX1	A¹	0XXX XXXX	A¹	0XXX XXXX	$\overline{A}$	P

Adreßwort Lesen — Datenwort ²) — Datenwort ²)

II. Schreibvorgang (z.B. »Tag« und »26« für Tag und »10« für Monat):

S	1101 0 A1 A0 0	A¹	0000 0010	A¹	0010 0110	A¹	0001 0000	A¹	P

Adreßbyte Schreiben — Subadreßwort »Tag« — 1. Datenwort ²) — 2. Datenwort ²)

Bild 2.51: Beispiele für Lesen aus dem und Schreiben in den PCB 8573

in diesem Fall also vom Computer – eine Stoppbedingung folgt. Die beschriebenen Verhältnisse wurden nocheinmal in Bild 2.51 grafisch dargestellt. Wenn Sie acht Antwortbyte angefordert haben, muß zweimal die gleiche Viererkombination ausgegeben worden sein.

Sehen Sie sich nun Teil II der Abbildung 2.51 an. Es handelt sich um den Schreibvorgang in den Baustein, also das Stellen der Uhr. In unserem Beispiel wollen wir das aktuelle Datum verändern. Dazu müssen wir zunächst wieder den PCF 8573 adressieren. Wir senden wieder

`11010000`, also hexadezimal `D0`,

um im nächsten Byte, der Subadresse, den Registerzeiger auf Zeitzähler-Tag zu setzen. Aus Tabelle 2.10 lesen wir ab

`00000010`, also hexadezimal `2`

(dritte Zeile der Tabelle). Im Gegensatz zum Lesevorgang folgen aber nun weitere Bytes, nämlich die neuen Inhalte der fol-

genden Register. Auch hier werden die Bits B0 und B1 des Registerzeigers mit jedem Acknowledge-Impuls automatisch um eins erhöht.

In dem abgebildeten Beispiel wird das Datum auf den 26.10. gesetzt. Da keine Antwortbytes zu berücksichtigen sind, lautet der entsprechende Befehl

```
I²C 0 D0 2 26 10
```

Der zweite Teil der Tabelle 2.10 betrifft die einzelnen Marken. Für X kann entweder eine 1 oder eine 0 eingesetzt werden. Wie Tabelle 2.11 zeigt, sind alle Marken gemeinsam in einem Byte untergebracht. Sie werden ganz analog zu den schon behandelten Registern gelesen, mit der Bytefolge:

`11010000` (hexadezimal D0) : Bausteinadresse zum Schreiben
`00010000` (hexadezimal 10) : Subadresse »Lese Zustand«

und anschließend:

`11010001` (hexadezimal D1) : Bausteinadresse zum Lesen.

Für die Handlingroutine bedeutet das die beiden Aufrufe

```
I²C 0 D0 10
I²C 1 D1
```

Die Kodierung der Befehle zeigt ebenfalls Tabelle 2.10.

1 Uhrenbaustein PCF 8573 (= PCB 8573)
2 Treiber 7407
2 IC-Sockel 14-polig
1 IC-Sockel 16-polig
1 Transistor BC558
1 Transistor BC107
1 Diode 1N4148
1 Widerstand 1 Megaohm
1 Widerstand 680 Kiloohm
1 Widerstand 560 Kiloohm
1 Widerstand 15 Kiloohm, 10 Kiloohm
3 Widerstände 3,3 Kiloohm
1 Widerstand (Wert siehe Text, z.B. 120 Ohm)
2 Kondensatoren 0,1 Mikrofarad
1 Trimmkondensator 22 Pikofarad
1 Miniaturquarz 32,768kHz
1 25-poliger Min-D-Stecker, für Amiga 1000 weiblich, sonst männlich
1 Nickel/Cadmium-Akku, 1,2 Volt
1 einseitige Platine nach Bild 2.52
1 Pfostensteckerleiste, einreihig 6pol; einreihig 12pol
2 Lötnägel

Tabelle 2.12: Stückliste für Druckertreiber, I²C-Interface und Uhr

2.6.7 Platine nicht nur für den



Angabe der Stunde und MM schließlich für die Minuten. Der Befehl

```
SETClock -s 16-AUG-89 19:34
```

stellt die Uhr auf den 16.08.1989, 19 Uhr 34. Kopieren Sie das Programm SETClock am besten ins C-Verzeichnis Ihrer Boot-Diskette und fügen Sie in der Startup-Sequence im s-Ordner das Kommando SETClock –l –y89 ein. Die interne Uhr wird dann beim Systemstart selbsttätig auf den Stand der I²C-Uhr gebracht. Hinter –y muß natürlich die aktuelle Jahreszahl folgen.

Mit der Option –d (Display) erfolgt eine Anzeige der I²C-Uhrzeit ohne Stellen der Systemzeit.

2.6.8 Weitere Bausteine für den I²C-Bus

Mit dem beschriebenen Schnittstellen-Zusatz und der Handlingroutine findet die I²C-Bus-Familie jetzt problemlos am Amiga Anschluß. Vielseitige Erweiterungen sind ohne großen Aufwand realisierbar. Tabelle 2.13 enthält einige für Erweiterungen interessante Bausteine.

PCF 8574	8 Bit I/O Baustein
PCF 8575	30 Bit LED-Treiber
PCB 8582	256 x 8 Bit EEPROM
PCF 8591	Mehrkanal Analog/ Digital- und Digital/ Analogwandler, 8 Bit
SAA 1300	5-fach Leistungstreiber
MAE 8000	Sprachsynthesizer
TEA 6300	Sound Fader Control Circuit
SAB 5230/5240	Videotext-Decoder

Tabelle 2.13: Einige interessante I²C-Bausteine von VALVO

Die Bauteilreihe wird laufend ergänzt und erweitert. Ein interessanter Baustein, besonders für Musikfreunde, ist der »Sound Fader Control Circuit« TEA 6300, mit dem – ohne mechanische Teile – allein über den I²C-Bus aus drei Stereo-Signalquellen eine ausgewählt und in Klang und Lautstärke beeinflußt werden kann. Durch vier Ausgänge läßt sich auch aus dem Amiga-Tonsignal programmierbarer Raumklang erzeugen.

Alle I²C-Bausteine sind mit Hilfe der Handlingroutine ähnlich einfach programmierbar wie die vorgestellte Uhren-/ Kalender-Einheit.

2.7 Analog/Digital-Wandlung

Die Fähigkeit, Naturklänge originalgetreu auszugeben, wurde dem Amiga mit seinen vier Audio-Wandlern bereits in die Wiege gelegt. Die Vorspänne vieler Programme machen davon ausgiebig Gebrauch. Hier soll nun das Gegenstück vorgestellt werden: Ein Zusatz, der Spannungen sehr schnell in Bitkombinationen umwandelt. Auf der Platine ist außerdem ein empfindlicher Tonfrequenzvorverstärker enthalten.

2.7.1 Die sukzessive Approximation

Lassen Sie sich von diesem Zungenbrecher nicht einschüchtern. Das Verfahren an sich ist ganz einfach. Bevor wir aber in die Materie einsteigen, sei noch darauf hingewiesen, daß auch die digitalen Meßwert-

aufnehmer aus Kapitel 2.4.9 (Kodierte Lichtschrankenscheibe) als Analog-Digital-Umsetzer aufgefaßt werden können, denn sie erfassen eine analoge Größe – zum Beispiel Winkel, Weg, Drehzahl – und geben ein digitales Signal ab, sind also bereits mechanisch-elektrische Analog-Digital-Umsetzer. In den nachfolgend beschriebenen Schaltungen wird demgegenüber davon ausgegangen, daß ein analoges Signal in Form einer Spannung vorliegt. Es handelt sich damit gleichzeitig um sogenannte Digital-Voltmeter, also Geräte zum Erfassen analoger Spannungen mit digitaler Anzeige. Doch zurück zur eigentlichen Umsetzung.

Analog/Digital-Wandler, wie sie sowohl im Audio- als auch im Video-Teil des Amiga enthalten sind, lassen sich relativ einfach realisieren. An dieser Stelle sei bereits der Verweis auf die Video-Wandler in Kapitel 4.1.4 erlaubt. Dort werden die zugehörigen Spannungen aller gesetzten Bits einfach analog addiert, und schon steht der Wert fest. So leicht ist die Sache hier nicht, denn zwischen den mit fester Bitzahl erreichbaren Werten liegen immer unendlich viele Zwischengrößen. Man ermittelt daher den genauesten Digitalwert, indem man sich ihm stufenweise (sukzessive) annähert (approximiert). Die Eingangsspannung wird mit einer festen Referenzspannung verglichen und so nacheinander für jedes Bit entschieden, ob es gesetzt werden muß oder nicht. Vergleichbar ist diese Annäherung etwa der Suche in einem Telefonbuch. Sie schauen sich die Buchstabenkombination oben auf der Seite an und entscheiden, ob Sie noch eine Seite weiterblättern müssen oder nicht.

Die sukzessiven Approximation kommt mit relativ wenig Aufwand aus und ist recht schnell. Umsetzer nach diesem Verfahren werden als komplette ICs angeboten, so auch der ZN 427 von Ferranti, der im vorgestellten Wandler zum Einsatz kommen soll.

2.7.2 Der ZN 427 als A/D-Wandler

Auf den ersten Blick ist der ZN 427 ein ganz gewöhnliches IC. Tatsächlich handelt es sich jedoch um einen sehr schnellen Wandler, der bei einer Taktfrequenz von 1MHz weniger als 10 Mikrosekunden für die Ermittlung eines vollständigen 8-Bit-Wertes benötigt. Er enthält eine präzise, temperaturstabile Referenzspannungsquelle, ist sowohl TTL- als auch CMOS-kompatibel und besitzt als besonderen Clou eine komplette Mikrocomputer-Schnittstelle. Allerdings benötigt er zum Betrieb eine zusätzliche negative Hilfsspannung.

Schauen wir uns zunächst das Blockschaltbild des ZN 427 (Bild 2.55) näher an. Pin 6 ist der Eingang für die analoge Meßspannung. Zwischen Pin 8 und 9 (Masse) liegt die Referenzspannungsquelle, die etwa so anzusehen ist, wie eine Zenerdiode, und mit einem Widerstand nach Pin 10 (+5V) belastet werden muß. Über Pin 7 kann dem IC eine externe Vergleichsspannung zugeführt werden. Im Normalfall sind beide Anschlüsse kurzgeschlossen. Der Meßbereich beträgt dann ±2,56 Volt.

Bei einer negativen Flanke an seinem Pin 4 ($\overline{WR}$) startet der ZN 427 die Wandlung, indem er Pin 1 ($\overline{BUSY}$) auf LOW zieht

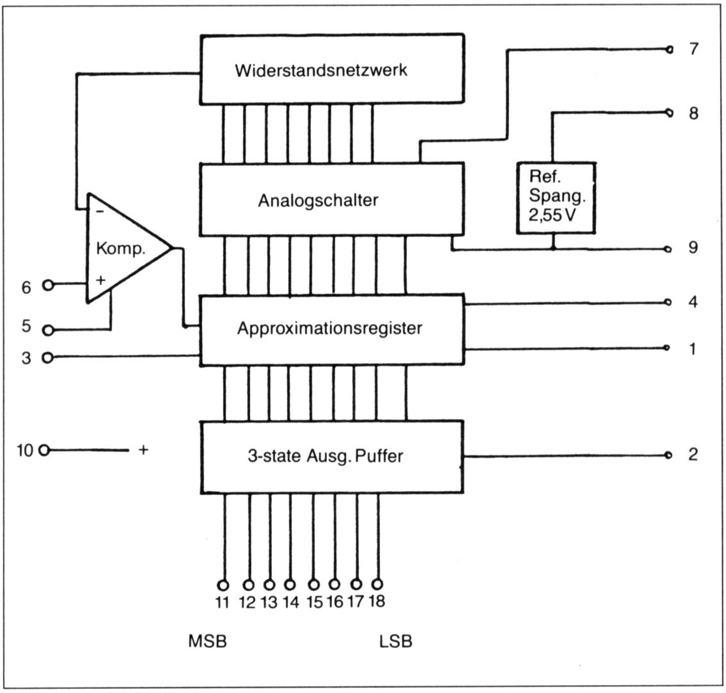


Bild 2.55: Blockschaltbild des A/D-Wandlers ZN 427 von Ferranti

und intern zunächst alle Datenbits auf einen Anfangszustand setzt. Sobald $\overline{WR}$ wieder nach HIGH geht, erfolgt nacheinander bei jedem Taktimpuls die Ermittlung eines Bits, wobei das höchstwertige zuerst gültig ist. Nach neun Taktzyklen geht $\overline{BUSY}$ wieder nach HIGH und zeigt damit das Ende der Wandlung an.

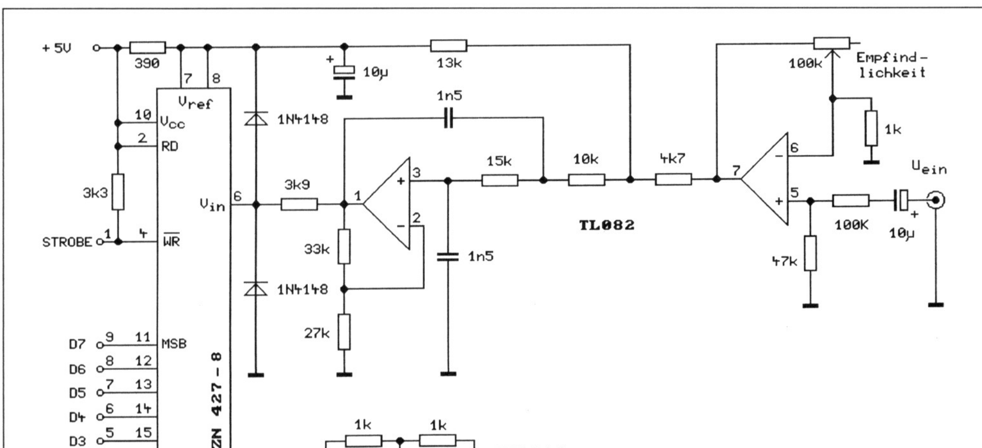
Beim Auslesen der Daten verhält sich der ZN 427 ähnlich wie ein EPROM. Seine Datenausgänge besitzen nämlich Tristate-Charakteristik, das heißt, sie lassen sich über Pin 2 (RD) ein- und ausschalten. Dieser Anschluß entspricht dem EPROM-Eingang $\overline{OE}$, verhält sich aber hier HIGH-aktiv. Im Gegensatz zu EPROMs sind die Datenausgänge also bei HIGH durchgeschaltet und bei LOW in einem hochohmigen Zustand, der den Datenbus nicht belastet. So könnte der ZN 427 mit sehr wenigen zusätzlichen Gattern auch direkt an den Systembus des Amiga angeschlossen werden. Hier erfolgt die Ansteuerung jedoch über den Parallelport, weil es bereits viele Programme gibt, die mit einem dort angeschlossenen Digitizer zusammenarbeiten.

2.7.3 Die Wandlerkarte am Parallelport

Bild 2.56 zeigt den Schaltplan einer A/D-Wandlerkarte mit dem ZN 427 für den Parallelport des Amiga. Zum Starten der Wandlung wird der STROBE-Impuls eingesetzt. Er ist direkt mit dem $\overline{WR}$-Eingang des ZN 427 verbunden. So wird

bereits beim Lesen des Wandlungsergebnisses selbsttätig eine neue Wandlung gestartet. Dadurch arbeitet der Digitizer mit fast allen Digitizer-Programmen. Wichtig für eine einwandfreie Funktion ist der Pull-up-Widerstand. Das Ende der Wandlung wird mit HIGH an BUSY gemeldet. Wird der Port vor der vollständig abgeschlossenen Wandlung gelesen, dann sind die unteren Bits noch gesetzt. Der Wert liegt also schon in der richtigen Größenordnung, ist aber entsprechend ungenau.

Der ZN 427 ist schnell genug zur Abtastung von Tonfrequenzsignalen. Daher wurde gleich ein entsprechender Verstärker auf der Platine untergebracht. Seine relativ aufwendige Schaltung dient nicht nur zur Anpassung des Pegels, sondern erfüllt gleichzeitig noch eine weitere wichtige Aufgabe. Beim Wandlungsverfahren der sukzessiven Approximation darf sich die Eingangsspannung während der Umsetzung nicht ändern, da sonst falsche Ergebnisse ermittelt würden. Pfeifen und Zwitschern wären die Folge. Daher ist ein aktiver Tiefpaß enthalten, der alle Frequenzen oberhalb von etwa 10 kHz unterdrückt. Vor dem Tiefpaß liegt noch ein Vorverstärker, dessen Empfindlichkeit in einem weiten Bereich mit einem Poti ein-



stellbar ist. Auf diese Weise sind sogar Mikrofone direkt anschließbar. Die Referenzspannung des Wandlers wird über einen Spannungsteiler in die Filterstufe eingespeist. So stellt sich die Anordnung selbsttätig auf einen Ruhewert in der Mitte des 8-Bit-Bereiches ein, und es ist gewährleistet, daß beide Halbwellen des Tonsignals gleich gut verarbeitet werden. Der maximale Eingangspegel liegt damit bei etwa 3Vss, also bei etwa 1Veff.

Die nötige negative Hilfsspannung von –5 Volt für Operationsverstärker und Wandler wird mit einem Spannungsumsetzer vom Typ ICL 7660 erzeugt.

Seinen Takt erhält der ZN 427 von einem Quarzoszillator. Diese Ausführung wurde einem RC-Glied vorgezogen, da nur so die volle Wandelgeschwindigkeit ausgenutzt

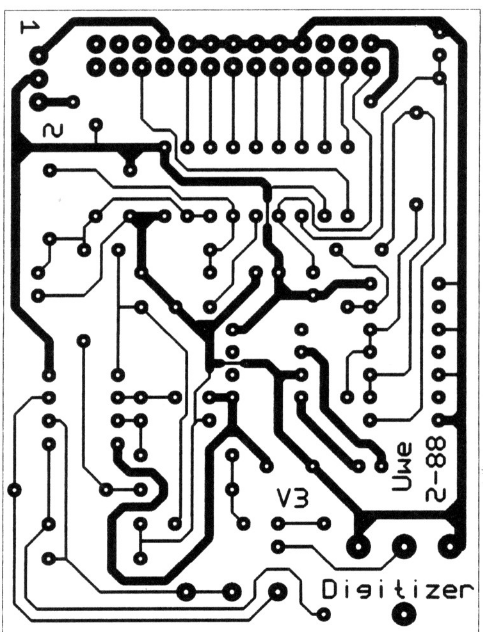


Bild 2.57: Platinenvorlage zum Amiga-Digitizer

werden kann. Es ist auch möglich, statt der Anordnung mit einem 74LS14 einen fertigen Oszillatorbaustein einzusetzen. Er paßt in den Sockel des 74LS14. Dabei entfallen die beiden Widerstände, der Quarz und der Kondensator.

2.7.4 Aufbau des A/D-Wandlers

Bild 2.57 zeigt die Platine des Analog/Digital-Wandlers. Obwohl ihr Schaltplan (Bild 2.56) recht unscheinbar aussieht, ist sie dicht bepackt und enthält vor allem viele Leiterbahnen. Richten Sie sich beim Herstellen der Platine und beim Bestücken nach den Hinweisen im Anhang A. Tabelle 2.14 gibt die benötigten Bauteile an, Bild 2.58 enthält den Bestückungsplan.

Die fertige Platine ist auf Foto 2.3 zu sehen. Die NF-Zuleitung sollte abgeschirmt ausgeführt sein, wobei das Geflecht an Masse liegen muß. Die Verbindung zum Rechner erfolgt über ein Flachbandkabel, das auf die Pfostensteckerleiste der Platine aufgesteckt wird. Es kann mit dem Sub-D-Stecker für den Rechner 1:1 verbunden werden. Lediglich die überschüssi-

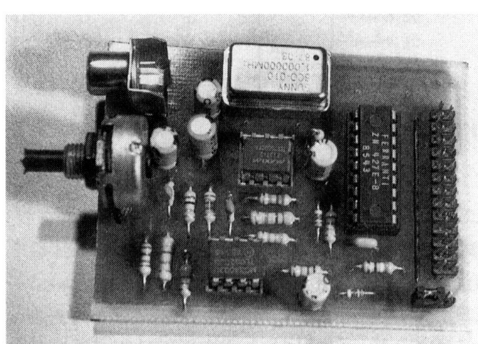


Foto 2.3: Digitizer

1	A/D-Wandler ZN 427 E-8
1	Doppel-Operationsverstärker TL 082
1	DC/DC-Wandler ICL 7660
1	IC-Sockel 18-pol
1	IC-Sockel 14-pol
2	IC-Sockel 8-pol
2	Dioden 1N4148
1	Widerstand 100 Kiloohm
1	Widerstand 82 Kiloohm
1	Widerstand 47 Kiloohm
1	Widerstand 82 Kiloohm
1	Widerstand 33 Kiloohm
1	Widerstand 27 Kiloohm
1	Widerstand 15 Kiloohm
1	Widerstand 13 Kiloohm
1	Widerstand 10 Kiloohm
1	Widerstand 3,3 Kiloohm
1	Widerstand 4,7 Kiloohm
1	Widerstand 3,9 Kiloohm
1	Widerstand 1 Kiloohm
1	Widerstand 390 Ohm
1	Potentiometer 100 Kiloohm, logarithmisch
5	Elektrolytkondensatoren 10 Mikrofarad / 16 Volt, radial
1	Kondensator 0,1 Mikrofarad, Keramik
2	Kondensatoren 1,5 Nanofarad
1	Cynch-Buchse für Printmontage
1	Stiftleiste 3pol für Jumper
1	doppelseitige Platine nach Bild 2.57
wahlweise	
1	Quarzoszillator 1 MHz
oder	
1	Logik-Baustein 74LS14
1	Quarz 1 MHz, kleine Bauform
2	Widerstände 1 Kiloohm
1	Kondensator 680 Pikofarad

Tabelle 2.14: Einkaufszettel für den A/D-Wandler

ge 26. Ader bleibt am 25-poligen Stecker unbeschaltet. Auf der Platine kann mit einem Jumper bestimmt werden, ob die Stromversorgung über einen Amiga 1000 oder über den Amiga 500/2000 geschehen soll.

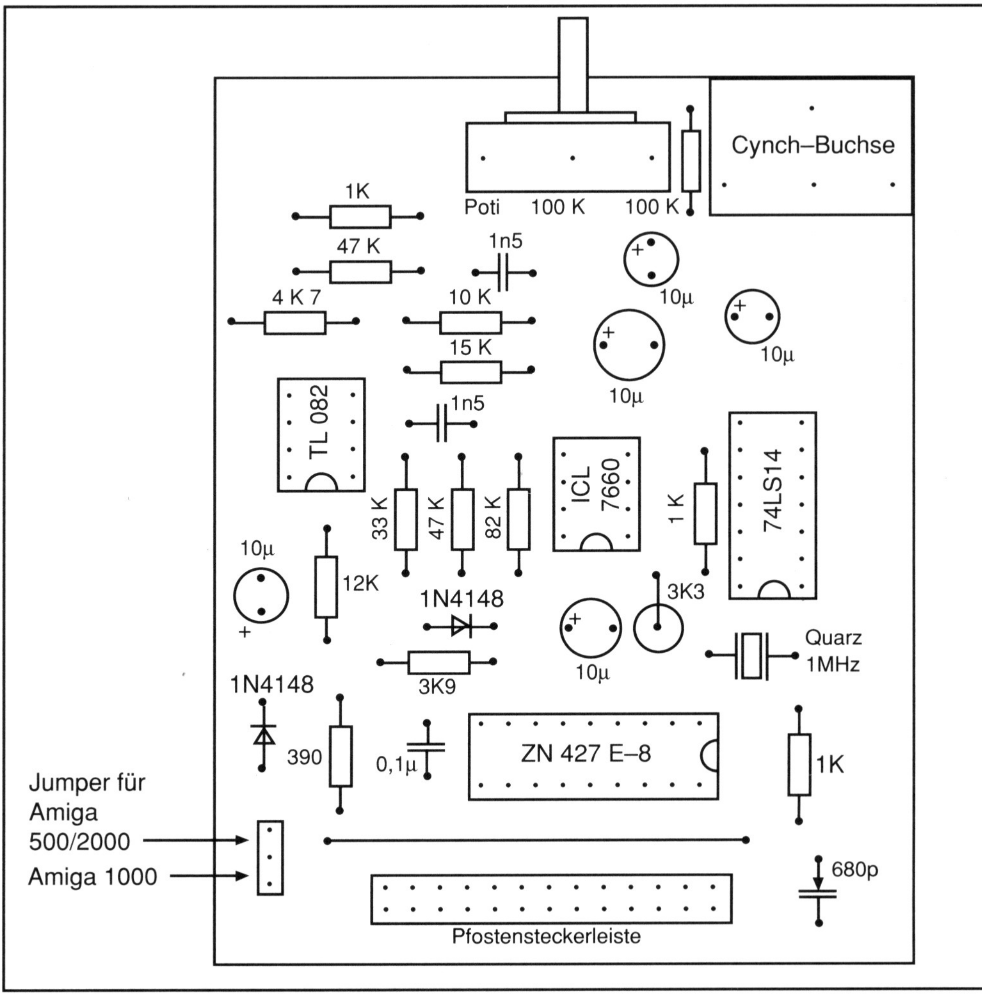


Bild 2.58: Bestückung des Digitizers

2.7.5 Das gehört zum guten Ton

Die Audio-Ausgabe wird im DMA-Kapitel 3.7 genauer erläutert. Auf der Diskette zum Buch befindet sich ein Digitizer-Programm von Christian Wolf, das in ähnlicher Version bereits in der 68000'er 6/87 veröffentlicht, und für das hier vorgestellte Gerät angepaßt wurde. Dort können Sie auf Seite 96 eine genaue Bedienungsanleitung nachschlagen. Hier nur das Wichtigste in Kürze.

Das Programm wurde einer Vierspur-Tonbandmaschine nachempfunden. Jeder Spur ist ein Amiga-Tonkanal zugeordnet. Sie lassen sich durch Anklicken von ON/OFF aus der Leiste am oberen Bildschirmrand einzeln oder gemeinsam aktivieren. ONCE/LOOP wählt aus, ob die

Spur nur einmal (ONCE) oder fortlaufend (LOOP) abgespielt werden soll.

Im Grafikfenster erscheint der Amplitudenverlauf des gesampelten Sounds. Die nun folgenden Parameter lassen sich durch Eingeben von Werten über die Tastatur oder nach Anklicken mittels des angedeuteten Schiebereglers verändern. »Start« legt fest, ab welcher Position der

»Future Sound« beinhaltet zusätzlich Informationen über Länge der Datei und Wiedergabegeschwindigkeit, und »Data« speichert ausschließlich die digitalisierten Daten. Bei »Load« wird das Format selbständig erkannt.

Unter den übrigen Menüpunkten finden Sie Funktionen zur Manipulation der gerade angewählten Spur (Track). »Clear«

Information	Bytes	Bedeutung
OneShotHiSamples	4	Anzahl der Bytes für Anschlag
RepeatHiSamples	4	Anzahl der Bytes für Klang
SamplesPerHiCycle	4	gerade Anzahl Bytes pro Schwingung
SamplesPerSec	2	Sample-Datenrate
ctOktaves	1	Anzahl der Oktaven im Chunk BODY
Pack	1	0: ungepackte Daten in BODY
Volume	4	Lautstärke (Normalwert -00010000)

Damit eröffnen sich eine ganze Reihe von Möglichkeiten. Grundsätzlich lassen sich zwei Verwendungsarten unterscheiden:

Beim »One-Shot-Sound« handelt es sich um einen einmaligen Effekt. Er wird mit einer bestimmten Wandlungsrate aufgenommen und später auch in dieser Form wiedergegeben. Benötigt wird hierzu der Wert SamplesPerSec für die Datenrate, sowie die Angabe der OneShotHiSamples und RepeatHiSamples für die Anzahl der Daten. Ist SamplesPerHiCycle unbekannt, wird eine 0 eingesetzt. Die Anzahl der Oktaven ist hier 1.

Ein »Musical Instrument« bietet im Vergleich zu reinen Effekten wesentlich mehr Komfort. Hier wird der Klang eines Instruments in verschiedenen Abschnitten digitalisiert. Beim Niederdrücken einer Taste muß später zunächst der erste Teil der Sounddaten (zum Beispiel das Anschlagen einer Gitarrensaite) einmalig gespielt werden, der Rest so lange, wie die Taste gedrückt bleibt. Der erste Teil heißt One-Shot-Part, der zweite Repeat-Part. Soll ein Klang als Musikinstrument wiedergegeben werden, läßt er sich in mehreren Oktaven ablegen, um eine naturgetreuere Wiedergabe in allen Tonlagen zu erreichen.

Die Angaben im Voice Header beziehen sich immer auf die höchste gespeicherte Oktave. Wurde in ctOktaves die Anzahl der im Chunk BODY gespeicherten Daten mit einem Wert größer 1 angegeben, dann müssen die Daten für die übrigen Oktaven, also One-Shot-Part und Repeat-Part, auf die der ersten folgen.

Von Oktave zu Oktave wird dabei doppelt soviel Speicher benötigt. Unter SamplesPerHiCycle muß die Anzahl der Bytes pro Schwingung in der höchsten Oktave angegeben werden.

Pack ist normalerweise 0. Dies besagt, daß die Daten ungepackt vorliegen. Bei 1 sind die Daten in BODY nach dem Fibonacci-Delta-Algorithmus gepackt. Dies kommt jedoch nur selten vor.

Für die Lautstärke wurde mit vier Byte ein sehr fein einstellbarer Bereich reserviert. Ihre Normaleinstellung ist $00010000. Sie läßt sich außerdem über die Daten in den Chunks ATAK und RLSE beeinflussen. Darauf soll jedoch in diesem Rahmen nicht eingegangen werden.

Der Chunk BODY enthält die gesampelten Sound-Daten. Dabei folgen die Informationen der im Header angegebenen Anzahl von Oktaven direkt aufeinander, sie bestehen jeweils aus einem OneShot- und einem Repeat-Teil. Ihre Länge verdoppelt sich von Oktave zu Oktave. Die Angaben im Header VHDR beziehen sich jedoch immer auf die höchste Oktave.

Die gesampelten Daten werden vorzeichenbehaftet abgespeichert. Bemerkenswert dabei ist, daß ihr höchstwertiges Bit gegenüber der normalen binären Zahlendarstellung invertiert ist. Der kleinste Amplitudenwert wird also durch %10000000 repräsentiert, der größte durch %01111111.

In den übrigen Chunks stehen für den Klang völlig unerhebliche Werte. Sie dienen vorwiegend der Information und bestehen in der Regel aus ASCII-Zeichen.

Hier ein Beispiel für eine IFF-Sound-Datei. Es handelt sich um den Anfang der Klangbeschreibung einer Baßgitarre.

Nach der allgemeinen IFF-Spezifikation FORM gefolgt von der Länge des Konstrukts ($348C) und dem FORM-Typ (8SVX) beginnt sofort der Header-Chunk VHDR mit einer Länge von 20 (= $14) Byte. Darin wird zunächst die Länge des One-Shot-Parts mit $1A8 Byte und des Repeat-Parts mit 8 Byte festgelegt. Die Byteanzahl pro Schwingung beträgt 4 und die Anzahl der Bytes pro Sekunde $291E, also 10625 Hz. Das nächste Byte kündigt 5 Oktaven im Chunk BODY an. Danach steht eine 0. Die Daten in BODY sind also nicht komprimiert. Schließlich steht noch die Lautstärke auf dem Standardwert $00010000.

Im sich anschließenden Chunk NAME ist der Sound-Name mit »bass guitar« angegeben, gefolgt von einem Füllbyte.

Dann beginnt der letzte Chunk BODY. Er hat eine Länge von $3450 und wurde hier aus Platzgründen nicht vollständig abgedruckt.

```
0000: 46 4F 52 4D 00 00 34 8C 38 53 56 58 56 48 44 52  FORM..4.8SVXVHDR
0010: 00 00 00 14 00 00 01 A8 00 00 00 08 00 00 00 04  ................
0020: 29 1E 05 00 00 01 00 00 4E 41 4D 45 00 00 00 0B  ).......NAME....
0030: 62 61 73 73 20 67 75 69 74 61 72 00 42 4F 44 59  bass guitar.BODY
0040: 00 00 34 50 CB 29 E7 B8 6F 6B 1E E8 C9 49 DC AB  ..4P.)..ok...I..
0050: 2B 1F 0A 00 E2 3D D7 B3 1E 1A 0C FC E4 3C D8 B7  +....=.......<..
0060: 1D 1A 0B FB E4 3D D7 B9 1C 1A 0C F9 E6 3C D6 BC  .....=.......<..
0070: 1B 1A 0D F9 E6 3C D3 BE 1A 1A 0E F8 E7 3C D2 BE  .....<.......<..
...
```

2.8 EPROM-Programmierer

Wir kommen jetzt zu einem universellen EPROM-Programmier-Zusatz. Er eignet sich für die Typen 2732A, P2764, 2764A, 27128, 27256 und 27512. Dabei wird die Anpassung der jeweiligen Pinbelegung voll programmgesteuert vorgenommen. Es sind jedoch auch die Daten anderer EPROMs einstellbar und mit einem Adapterstecker können auch beliebige andere EPROMs programmiert werden.

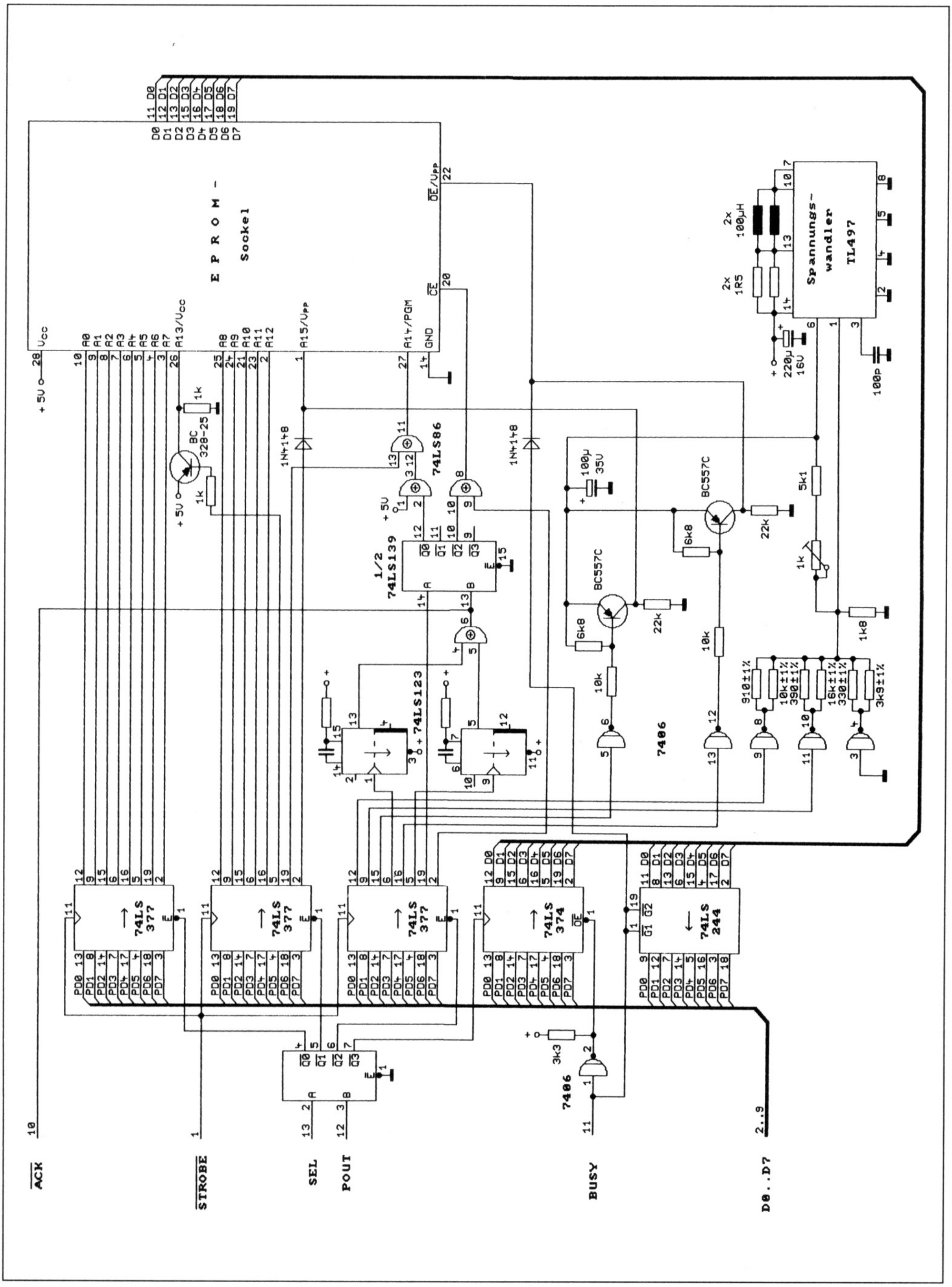


Bild 2.59: Schaltplan des EPROM-Brenners

2.8.1 Die EPROMmer-Hardware

Bild 2.59 zeigt den Schaltplan des EPROM-Brenners. Er wird an den Parallelport des Amiga angeschlossen, wobei alle Daten- und Steuerleitungen Verwendung finden. Ihre Funktion soll nun im einzelnen genauer beleuchtet werden.

2.8.1.1 Steuerung über Register

Vier Register dienen zur Aufnahme der nötigen Daten und Steuerinformationen. Jedes dieser Register kann unabhängig von den anderen über den Parallel-Port des Amiga beschrieben werden. Dazu muß es zunächst über die beiden Steuerleitungen POUT und SEL adressiert werden. Tabelle 2.15 zeigt die Zuordnung.

Signal Pin	SEL 13	POUT 12	Register
4	0	0	Adreßregister 1
5	0	1	Adreßregister 2
6	1	0	Steuerregister
7	1	1	Datenregister

Tabelle 2.15: Adressierung der Register im EPROMmer

Die Adreßregister 1 und 2 dienen zur Aufnahme der aktuellen EPROM-Adresse. Register 1 enthält den niederwertigen, Register 2 den höherwertigen Teil. Im Gegensatz zum Aufbau mit einem Adreßzähler können so einzelne EPROM-Bereiche direkt und damit wesentlich schneller adressiert werden.

Der Datenkanal hat in zweifacher Hinsicht eine Sonderstellung. Alle anderen Register werden durch den STROBE-Impuls selbständig getriggert. Ein neuer Wert braucht also nur auf den Port ausgegeben zu werden, damit die Daten am Ausgang sofort anstehen. Beim Datenregister werden die Port-Daten erst übernommen, wenn ein anderes Register selektiert wird. Diese Betriebsart wurde nötig, weil der Datenkanal auch lesbar sein muß. Mit BUSY läßt sich die Datenrichtung umschalten.

BUSY	Richtung
HIGH	Datenregister → EPROM
LOW	EPROM → Treiber → Port

Tabelle 2.16: Richtungssteuerung der Datenübertragung

Bei BUSY = HIGH sind die Datentreiber abgeschaltet und das Register legt seine Ausgänge auf die EPROM-Datenleitungen, bei BUSY = LOW kehren sich die Verhältnisse um. Allerdings sollte vor dem Einschalten dieses Zustands der Amiga-Port unbedingt bereits auf Eingabe programmiert worden sein! Andernfalls kann es zu Datenkonflikten kommen.

Mit der BUSY-Leitung wird gleichzeitig der EPROM-Eingang $\overline{OE}$ (Pin 22) beeinflußt, der beim Auslesen immer LOW, beim Programmieren jedoch HIGH sein, bzw. die Programmierspannung führen muß.

2.8.1.2 Umschaltung der EPROM-Typen

Leider hat jedes der unterstützten EPROMs eine leicht andere Pinzuordnung. Die Anschlußbelegung der einzelnen EPROMs können Sie im Anhang A nachschlagen. Tabelle 2.17 stellt in

Typ	2716	2732	P2764	2764A	27128	27256	27512
PGM-Spng.	25V	21V	21V	12,5V	21V	12,5V	12,5V
maximal	26V	22V	22V	13V	22V	13V	14V
an Pin	23	22	1	1	1	1	22
PGM-Pin	$\overline{CE}$	$\overline{CE}$	$\overline{PGM}$	$\overline{PGM}$	$\overline{PGM}$	$\overline{CE}$	$\overline{CE}$
Pin Nr.	20	20	27	27	27	20	20
Pulsform	HIGH	LOW	LOW	LOW	LOW	LOW	LOW
max.Länge	55ms	55ms	1+63ms	1+78,75ms	55ms	1+78,75ms	
Programmiermodus bei folgenden Pegeln:							
$\overline{CE}$ (20)	HIGHp	LOWp	LOW	LOW	LOW	LOWp	LOWp
$\overline{OE}$ (22)	HIGH	21V	HIGH	HIGH	HIGH	HIGH	12,5V
$\overline{PGM}$ (27)	–	–	LOWp	LOWp	LOWp	A14	A14
Pin 26	–	+5V	+5V	N.C.	A13	A13	A13
Verify-Modus bei folgenden Pegeln:							
$\overline{CE}$ (20)	LOW	LOW	LOW	LOW	LOW	HIGH	LOW
$\overline{OE}$ (22)	LOW	LOW	LOW	LOW	LOW	LOW	LOW
$\overline{PGM}$ (27)	–	–	HIGH	HIGH	HIGH	A14	A14
Pin 26	–	+5V	+5V	N.C.	A13	A13	A13

Tabelle 2.17: Gegenüberstellung verschiedener EPROMs

einer Übersicht die Hauptunterschiede der wichtigsten EPROM-Typen dar. Es wurden Angaben der Firma Intel zugrunde gelegt.

Die softwaremäßige Umschaltung der Kennwerte und Pinzuordnungen wird über ein Steuerregister vorgenommen, das die vielfältigen Möglichkeiten des EPROM-Programmierers kontrolliert. Jedem Bit ist eine spezielle Funktion zugeordnet. Einen Überblick gibt Tabelle 2.18.

Pin	Funktion
S 0	Wert der Programmierspannung
S 1	Wert der Programmierspannung
S 2	Programmierspannung an Pin 1
S 3	Impulsauslösung (Länge 1ms)
S 4	Programmierspannung an Pin 22
S 5	Impulsauslösung (Länge 10ms)
S 6	Impuls-Pin (0: Pin 20, 1: Pin 27)
S 7	Pegel Pin 20 ($\overline{CE}$)

Tabelle 2.18: Funktionen der Steuerregister-Bits

Die Steuerbits 0 und 1 erlauben die Auswahl einer von vier Spannungen, die dann über Schalttransistoren mittels Steuerbit 2 und 4 an den gewünschten EPROM-

S 0	S 1	Spannung
0	0	5,0 Volt
0	1	12,5 Volt
1	0	21,0 Volt
1	1	25,0 Volt

Tabelle 2.19: Spannungsauswahl mit Steuerbits S0 und S1

Anschluß gelegt werden kann. Tabelle 2.19 zeigt die einstellbaren Spannungen.

Ein weiterer Schalttransistor versorgt noch Pin 26, der bei EPROMs mit 24-poligem Gehäuse +5 Volt Versorgungsspannung, bei 28-poligen Ausführungen jedoch A13 anlegen muß. Bei LOW schaltet der Transistor +5 Volt mit der benötigten Stromstärke an Pin 26 des EPROM-Sockels durch. Andernfalls liegt der Anschluß über einen Widerstand auf LOW.

Neben einer speziellen Spannung wird für den Programmiervorgang noch ein Programmierimpuls benötigt. Die Schaltung bietet dazu zwei verschiedene Impulslängen an, die wahlweise verwendet werden können. Eine positive Flanke, also ein LOW → HIGH-Wechsel an Steuerpin 3 oder 5 startet den Impuls. Durch weitere positive Triggerflanken während der Schaltzeit kann der Impuls auch beliebig verlängert werden.

Steuerbit 6 regelt die Auswahl des Pins, an dem der Programmierimpuls auftreten soll. Mit Ende des Impulses wird über die Portleitung $\overline{\text{ACK}}$ das Flag-Interrupt-Bit im Amiga-I/O-Baustein 8520-A (siehe Kapitel 1.2.2.1) gesetzt. Dies kann von der Software zyklisch abgefragt oder besser durch Auslösung eines Interrupts erkannt werden. Während der Dauer des Programmierimpulses leuchtet die über einen Transistor angesteuerte LED auf.

Steuerbit 6 schließlich läßt die Einstellung des logischen Pegels an Pin 20 ($\overline{\text{CE}}$) zu. Das ist nötig, weil dieser Anschluß beim 27256 im Verify-Modus HIGH sein muß.

2.8.1.3 Aufbau und Einsatz des Programmierers

Der Aufbau des EPROMmers erfolgt auf der doppelseitigen Platine nach dem Layout in Bild 2.60. Die Teile aus Tabelle 2.20 werden dabei nach dem Bestückungsplan in Bild 2.61 verlötet. Richten Sie sich bei der Herstellung der Platine und bei der Bestückung nach den Hinweisen in Anhang A. Vergessen Sie nicht die nötigen Durchkontaktierungen und verwenden Sie

1	Platine nach Bild 2.60
1	6-fach Open-Collector-Inverter 74LS06
1	4-fach Exor-Gatter 74LS86
1	2-fach Monoflop 74LS123
1	2-fach Demultiplexer 74LS139
1	8-Bit-Treiber 74LS244
1	8-Bit-Register 74LS374
3	8-Bit-Register 74LS377
1	Spannungswandler TL497 (Texas)

3	IC-Sockel, 14-pol
2	IC-Sockel, 16-pol
5	IC-Sockel, 20-pol
1	Transistor BC 328-25 o.ä.
1	Transistor BC 107 o.ä.
2	Transistoren BC557C o.ä.
2	Dioden 1N4148
1	Leuchtdiode, Farbe nach Wahl
1	Widerstand 330 Ohm +/- 1%
1	Widerstand 390 Ohm +/- 1%
1	Widerstand 910 Ohm +/- 1%
1	Widerstand 3,9 Kiloohm +/- 1%
1	Widerstand 10 Kiloohm +/- 1%
1	Widerstand 16 Kiloohm +/- 1%
2	Widerstände 33 Kiloohm +/- 1%
1	Widerstand 330 Ohm
2	Widerstände 1,5 Ohm
2	Widerstände 1 Kiloohm
1	Widerstand 1,8 Kiloohm
2	Widerstände 3,3 Kiloohm
1	Widerstand 5,1 Kiloohm
2	Widerstände 6,8 Kiloohm
3	Widerstände 10 Kiloohm
2	Widerstände 22 Kiloohm
1	Trimmpoti 1 Kiloohm
1	Kondensator 100 Pikofarad
1	Elektrolytkondensator 100 Mikrofarad / 35 Volt
1	Elektrolytkondensator 220 Mikrofarad / 16 Volt
1	Kondensator 68 Nanofarad
1	Kondensator 680 Nanofarad
2	Miniaturdrosseln 100 Mikrohenry
1	Nullkraftsockel, 28-pol (Textool)
1	Pfostensteckerleiste, 26-pol, doppelreihig
1	Pfostensteckerleiste, 3-pol, einreihig
1	Jumper (Kurzschlußbrücke für Pfostenstecker)

Tabelle 2.20: Bauteile für den EPROM-Programmierer

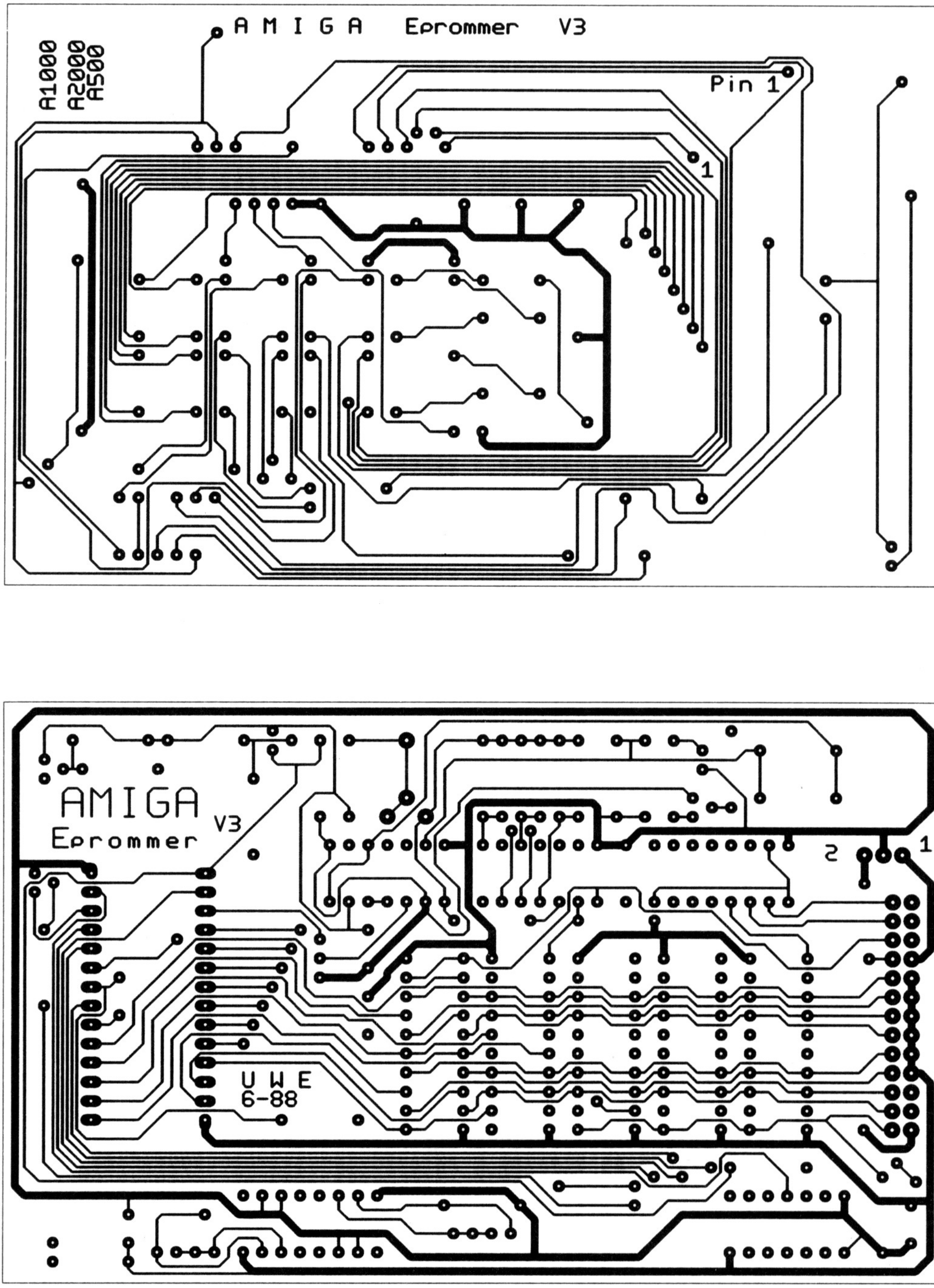


Bild 2.60: Layout für den EPROMmer

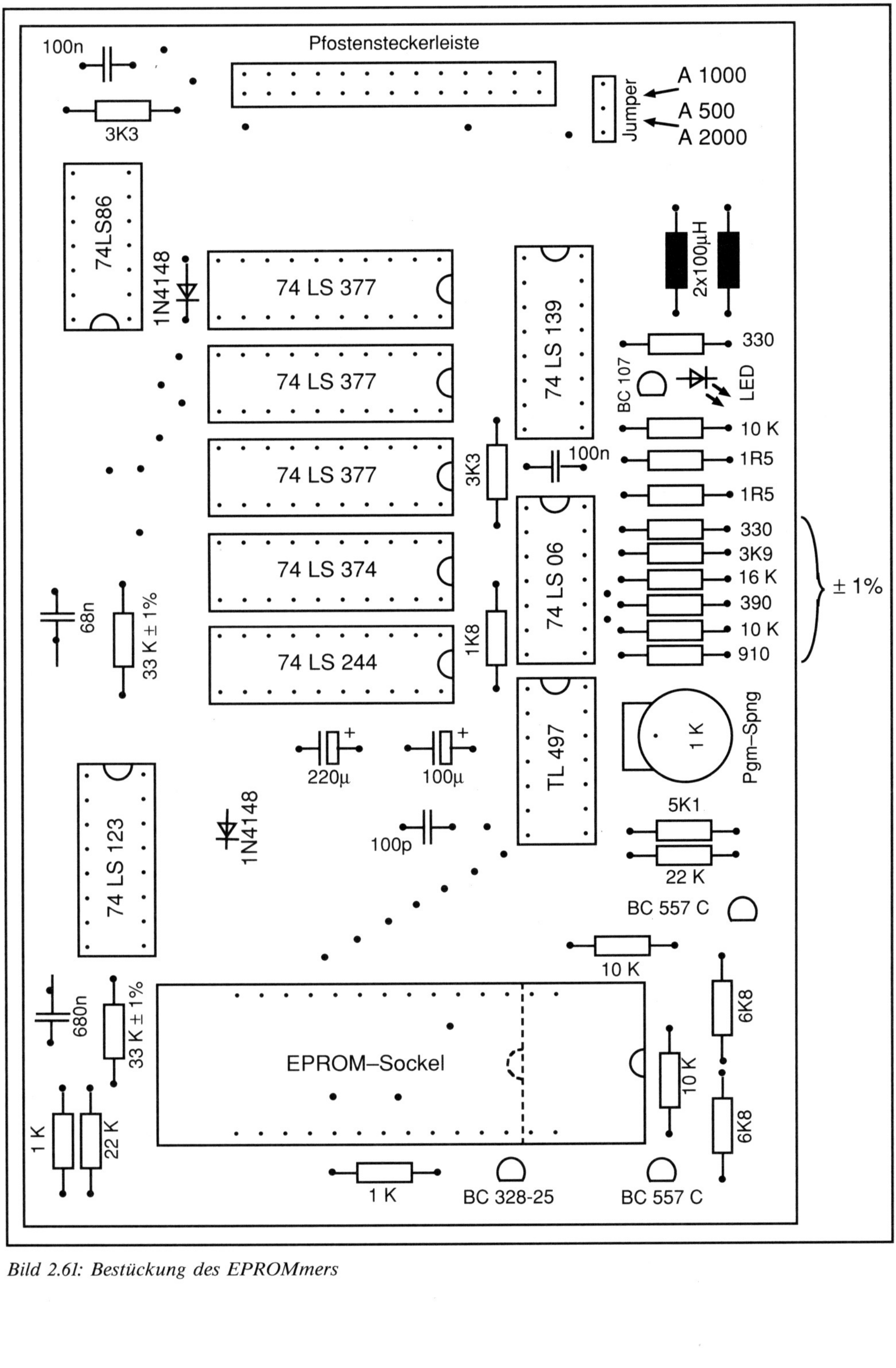


Bild 2.61: Bestückung des EPROMmers

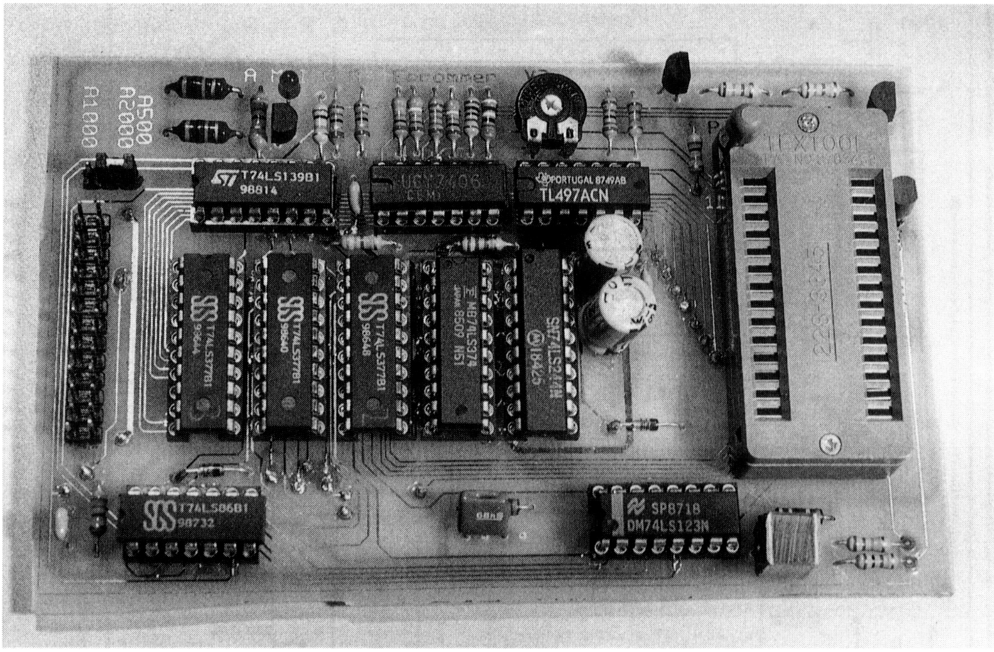


Foto 2.4: EPROMmer

Sockel mit gedrehten Beinchen, die auch von der Bestückungsseite her lötbar sind. Achten Sie bei der Reihenfolge der Bestückung darauf, daß die entsprechenden Lötstellen erreichbar bleiben.

Der Spannungswandler TL497 von Texas sollte bei allen gut sortierten Elektronikläden erhältlich sein. Anstelle der beiden Drosselspulen kann zur Not auch eine Netzentstördrossel mit 50 Mikrohenry verwendet werden.

Damit die Funktion der Hardware im Zusammenhang deutlich wird, folgt nun ein allgemeingültiges Beispiel für den Programmiervorgang. Zunächst müssen je nach EPROM die statischen Betriebswerte aus Tabelle 2.17 eingestellt werden. Das geschieht folgendermaßen:

- BUSY auf HIGH (Datenrichtung ist Amiga-Port → Datenregister → EPROM).
- Amiga-Port auf Ausgabe.
- Steuerregister selektieren (mit SEL und POUT).
- Bits für Programmierspannung und $\overline{\text{pin}}$ sowie Impulspin nach den Tabellen setzen. Die Impuls-Bits müssen dabei 0 sein.

Nun kann die eigentliche Programmierschleife beginnen. Sie läuft für jede EPROM-Adresse ungefähr folgendermaßen ab:

- Adreßregister 1 selektieren (mit SEL und POUT).

- Niederwertige Adreßhälfte auf den Port ausgeben. Sie wird automatisch im Adreßregister 1 gespeichert.
- Falls nötig ebenso Adreßregister 2 selektieren und höherwertige Adreßhälfte auf den Port ausgeben.
- Datenregister selektieren.
- Zu programmierendes Datenwort auf den Port ausgeben.
- Steuerregister selektieren. Erst damit wird der Port-Zustand ins Datenregister gespeichert und liegt wegen BUSY = HIGH auch an den EPROM-Datenleitungen an.
- Gewünschtes Impuls-Bit im Steuerregister setzten. Der Programmierimpuls wird damit ausgelöst.
- Das Bit gleich wieder zurücksetzen, Port auf Eingabe schalten und Impulsende durch Kontrolle von $\overline{\text{ACK}}$ abwarten.
- BUSY auf LOW setzen und damit Datenrichtung EPROM → Treiber → Port festlegen.
- Port lesen, Datenwort überprüfen und gegebenenfalls noch einen Impuls auslösen. (BUSY auf HIGH, Port auf Ausgabe, Impulsbit setzen und wieder zurücksetzen, BUSY auf LOW und Port auf Eingang.)
- Sonst BUSY wieder auf HIGH und mit der nächsten Adresse fortfahren.

2.8.2 EPROMmer-Software

Das auf der Diskette enthaltene Programm EPROMmer erlaubt die komfortable Benutzung der EPROMmer-Hardware. Die Programmierung erfolgt dabei interruptgesteuert im Hintergrund und benötigt extrem wenig Prozessorzeit. So werden andere ablaufende Tasks kaum gestört. Wegen der impulsgesteuerten Monoflops ist eine Zerstörung der EPROMs durch zu lange Programmierzeiten ausgeschlossen.

2.8.2.1 Programmstart

Das Programm EPROMmer befindet sich auf der beiliegenden Diskette. Nach dem Aufruf vom CLI aus oder durch Anklicken des Icons erscheint ein Programm-Fenster mit Namen EPROMmer, von dem aus die Funktionen steuer- und überwachbar sind. Dort befinden sich auf der linken Seite drei Felder zur Angabe von Start- und Stop-Adresse sowie für einen Offset. Nach Hineinklicken mit der Maus lassen sich die dort angegebenen Werte ändern. Reine Zahlen werden dezimal behandelt. Ein vorangestelltes $ kennzeichnet hexadezimale Werte. Alle Eingaben sind mit Return abzuschließen.

Die Vorgaben beziehen sich jeweils auf den aktuell ausgewählten EPROM-Typ, der darunter ständig angezeigt wird. »File« bezeichnet die Datei, mit der zuletzt gearbeitet wurde. Ganz unten im Fenster befindet sich noch eine Statuszeile, in der Rückmeldungen des Programms ausgegeben werden. Auf der rechten Seite des Programm-Fensters erscheinen die Felder »Start« und »Stop«, die wie Knöpfe zu

handhaben sind. Anklicken von »Start« mit der Maus startet die zuvor im Project-Menü ausgewählte Funktion, »Stop« beendet sie vorzeitig.

Aktiviert man das Programm-Fenster durch Hineinklicken, erscheint nach Drücken der rechten Maustaste eine Menüleiste am oberen Bildschirmrand, über die folgende Funktionen anwählbar sind:

Project	Mode	Type	Custom Mode
Open	Auto	2732A	Multiplier
Save	50ms Static	P2764	Max. Time
Read Eprom	Custom	2764A	
Program Eprom		27128	
Verify Eprom		27256	
Empty Test		27512	
Display		Custom	
Quit			

Die einzelnen Funktionen sind auch mit den im Menü jeweils angegebenen Tasten (bei gedrückter rechter Amiga-Taste) erreichbar.

2.8.2.2 Das Project-Menü

Dieses Menü enthält die Hauptfunktionen des EPROMmers. In den übrigen Menüs lassen sich lediglich Parameter wie EPROM-Größe und Programmierzeit einstellen. Der EPROMmer arbeitet mit einem Pufferspeicher, der die gelesenen oder zu programmierenden Daten enthält.

`Open`

Daten zum Programmieren eines EPROMs können in den Puffer geladen werden. Ist im Programmfenster ein Offset angegeben, werden entsprechend viele Bytes vom Anfang des Files überlesen und die folgenden Daten in den Pufferspeicher ab Adresse 0 übernommen.

Falls nach dem Laden des angegebenen Files noch Platz im Puffer freibleibt, erscheint die Warnung »Buffer not full!«

`Save`

Ermöglicht das Abspeichern von Daten aus dem Pufferspeicher. Der zu speichernde Bereich kann durch Angabe von Start- und Endadresse im Programmfenster festgelegt werden.

`Read Eprom`

Der Inhalt des im Programmierer eingesteckten EPROMs wird ausgelesen. Das Ausführen dieser Funktion erfolgt nach Anklicken des Start-Knopfes im Programm-Fenster. In der Statuszeile werden die bearbeiteten EPROM-Adressen angezeigt.

`Program Eprom`

Die im Speicher befindlichen Daten werden in das im Programmierer befindliche EPROM gebrannt. Das Ausführen dieser Funktion erfolgt nach Anklicken des Start-Knopfes im Programm-Fenster. Dabei sind die Angaben für Start- und Endadresse wirksam. Ein Offset wird nicht berücksichtigt. Während des Programmiervorgangs leuchtet die LED auf dem Programmierer. In der Statuszeile des Programm-Fensters erscheint die jeweils gerade bearbeitete EPROM-Adresse. Läßt sich eine Zelle nicht programmieren, führt das zu einer Fehlermeldung.

`Verify Eprom`

Die Daten im EPROM werden mit denen im Speicher verglichen. Das Ausführen dieser Funktion erfolgt nach Anklicken des Start-Knopfes im Programm-Fenster. Auch hier werden Start- und Endadresse, nicht jedoch ein Offset berücksichtigt. In der Statuszeile werden die bearbeiteten EPROM-Adressen angezeigt. Unterscheidet sich der EPROM-Inhalt von dem des Pufferspeichers, erscheint in der Statuszeile des Programm-Fensters die Meldung »Difference at Adr:« und die Angabe der ersten unterschiedlichen Adresse. Besteht kein Unterschied, endet die Funktion mit »Ready«.

`Empty Test`

Prüft, ob das im Programmierer befindliche EPROM leer ist. Die Inhalte aller Speicherzellen des gesamten EPROMs müssen dabei den Wert $FF besitzen. Der Test wird sofort nach Anwahl im Pull-down-Menü begonnen. Start- und Endadresse sowie der Offset werden nicht berücksichtigt. Die bearbeiteten Adressen werden in der Statuszeile sichtbar gemacht. Enthält mindestens eine Adresse einen von $FF verschiedenen Wert, erscheint die Anzeige »Eprom is not empty!«, andernfalls wird ausgegeben: »Eprom is empty!«

`Display`

Von Disk geladene oder aus einem EPROM ausgelesene Daten werden in einem eigenen Fenster auf dem Bildschirm angezeigt. Bei dieser Funktion kann das Ausgabeformat über ein eigenes Pull-down-Menü ausgewählt werden. Für nachfolgende EPROMmer-Operationen ist das Display-Fenster zunächst zu schließen.

`Quit`

Nach Anwahl dieses Menüpunktes wird das Programm verlassen.

2.8.2.3 Das Mode-Menü

Dieses Menü ist nur bei der Funktion Program EPROM von Bedeutung. Einer der drei Menüpunkte wird in der üblichen Weise aktiviert, indem er bei gedrückter rechter Maustaste angewählt und die Taste dann losgelassen wird.

`Auto`

Die Programmierung erfolgt nach einem speziellen, je nach ausgewähltem EPROM verschiedenen, Schnellprogrammieralgorithmus, wobei so lange immer wieder eine

feste Zeit programmiert wird, bis die Information im EPROM steht. Danach wird noch eine bestimmte Zeit nachgebrannt. Diese Programmiermethode spart besonders bei großen EPROM-Kapazitäten erheblich Programmierzeit ein und wird von den Chip-Herstellern empfohlen.

`50ms Static`

Jede Speicherzelle wird mit einer festen Zeit von 50ms programmiert. Das ist die Holzhammermethode für schwierige Fälle. Vorsicht: Die Gesamtprogrammierzeit wird sehr lang werden!

`Custom`

Auch hier wird die Schnellprogrammierung benutzt. Allerdings werden nicht die vorgegebenen Herstellerangaben benutzt, sondern die Zeiten und Faktoren lassen sich im Custom-Menü von Hand einstellen.

2.8.2.4 Das Type-Menü

Hier läßt sich bestimmen, mit welchem EPROM-Baustein gearbeitet werden soll. Der Programmierer unterstützt die Typen 2732A, P2764, 2764A, 27128, 27256 und 27512 direkt. Unter dem Menüpunkt Custom sind jedoch auch die Daten anderer EPROMs einstellbar. Nach der Auswahl eines EPROM-Typs wird ein entsprechend großer Puffer angelegt und gelöscht.

2.8.2.5 Das Custom-Mode-Menü

Dieses Menü ist nur von Bedeutung, wenn im Mode-Menü Custom ausgewählt wurde. Die Zahlen bestimmen die Programmierzeiten beim dort unter Auto erläuterten Schnellprogrammierverfahren.

`Multiplier`

Für die Berechnung der Nachbrennzeit kann der Faktor 2, 3, 4 oder 5 eingestellt werden.

`Max. Time`

Die Maximale Vorprogrammierzeit läßt sich auf 10ms, 15ms, 20ms, 25ms oder 30ms einstellen. Hat das EPROM danach nicht die Information übernommen, wird die Programmierung mit der Fehlermeldung »Eprom defekt« und Angabe der Adresse abgebrochen.

3

Die Amiga-Spezialchips

Den drei Spezial-Chips AGNUS (Adress Generator), DENISE (Display Encoder) und PAULA (Peripheral/Audio) verdankt der Amiga seine überragenden Eigenschaften. Überspitzt gesagt sind allein diese drei Bausteine bereits der Amiga. Alles andere ist nur Beiwerk. Das folgende Kapitel gibt einen Einblick in die Funktion dieser Amiga-Innereien und stellt anhand einiger Beispiele ihre Anwendung und Programmierung vor.

3.1 Triumvirat

Die drei Spezialchips im Amiga müssen grundsätzlich als eine Einheit gesehen werden, die lediglich aus fertigungstechnischen Gründen auf drei separarte Gehäuse aufgeteilt wurden. Dabei bedeutet vor allem die zur Herstellung der hochintegrierten Schaltungen benötigte Chip-Fläche einen sehr wichtigen Faktor. Die Bauteile werden in einem 4μm-HMOS-Verfahren gefertigt. Gegen die unvorstellbar winzigen Strukturen auf dem Chip wirken Staubteilchen bereits wie Felsbrocken. Trotz hochgradig staubfreier Fertigungsräume läßt sich nie ganz vermeiden, daß Verunreinigungen Störungen im Herstellungsprozeß hervorrufen. Ein einziges Staubkorn auf dem gesamten Chip macht ihn in der Regel bereits unbrauchbar.

Natürlich muß bei der Chip-Herstellung ein wirtschaftliches Verhältnis zwischen Ausbeute und Ausschuß angestrebt werden. Heute gilt allgemein durchschnittlich ein Defekt pro Quadratzentimeter als gerade noch vertretbar. Mit wachsender Fläche wird gleichzeitig auch der Ausschuß deutlich ansteigen und damit die Ausbeute sinken. Eine Auswirkung dieser Beschränkung ist beispielsweise die übliche Aufteilung von Prozessor und Arithmetik-Koprozessor auf zwei getrennte Bauteile, einfach weil beide Einheiten zusammen zu viel Chip-Fläche belegen würden.

Ähnlich sieht die Sache bei den drei Spezialchips im Amiga aus. Zwar lassen sich jedem Baustein gewisse Einzelfunktionen zuordnen, jedoch sind diese tatsächlich funktionell eng miteinander verzahnt. Das äußert sich schon darin, daß oftmals den Steuerbits einer einzigen Chip-Registeradresse Funktionsblöcke in unter-

Datenbus Bit 8	ein/aus	D8	1	48	D9	ein/aus	Datenbus Bit9
Datenbus Bit 7	ein/aus	D7	2	47	D10	ein/aus	Datenbus Bit10
Datenbus Bit 6	ein/aus	D6	3	46	D11	ein/aus	Datenbus Bit11
Datenbus Bit 5	ein/aus	D5	4	45	D12	ein/aus	Datenbus Bit12
Datenbus Bit 4	ein/aus	D4	5	44	D13	ein/aus	Datenbus Bit13
Datenbus Bit 3	ein/aus	D3	6	43	D14	ein/aus	Datenbus Bit14
Datenbus Bit 2	ein/aus	D2	7	42	D15	ein/aus	Datenbus Bit15
Datenbus Bit 1	ein/aus	D1	8	41	GND	---	Masse
Datenbus Bit 0	ein/aus	D0	9	42	-HSY	ein/aus	Horizontale-Synchronisation
Versorgungsspannung +5V	ein	VCC	10	39	-CSY	aus	Composite-Synchronimpuls
System Reset	ein	-RES	11	38	-VSY	ein/aus	Vertikal-Synchronimpuls
Interruptanforderung Ebene 3	aus	-INT3	12	37	-LP	ein	Lightpen
DMA Request Line	ein	DMAL	13	36	DRA8	aus	Dynamic RAM Address 8
Blitter Slowdown	ein	-BLS	14	35	DRA7	aus	Dynamic RAM Address 7
Data Bus Request	aus	-DBR	15	34	DRA6	aus	Dynamic RAM Address 6
AGNUS RAM Write	aus	-ARW	16	33	DRA5	aus	Dynamic RAM Address 5
Register Address 8	ein/aus	RGA8	17	32	DRA4	aus	Dynamic RAM Address 4
Register Address 7	ein/aus	RGA7	18	31	DRA3	aus	Dynamic RAM Address 3
Register Address 6	ein/aus	RGA6	19	30	DRA2	aus	Dynamic RAM Address 2
Register Address 5	ein/aus	RGA5	20	29	DRA1	aus	Dynamic RAM Address 1
Register Address 4	ein/aus	RGA4	21	28	DRA0	aus	Dynamic RAM Address 0
Register Address 3	ein/aus	RGA3	22	27	GND	---	Masse
Register Address 2	ein/aus	RGA2	23	26	CCKQ	ein	Color Clock Delay
Register Address 1	ein/aus	RGA1	24	25	CCK	ein	Color Clock

Bild 3.1: Pinbelegung von AGNUS

Funktion	Richtung	Signal	Pin	Pin	Signal	Richtung	Funktion
Datenbus Bit 6	ein/aus	D6	1	48	D7	ein/aus	Datenbus Bit 7
Datenbus Bit 5	ein/aus	D5	2	47	D8	ein/aus	Datenbus Bit 8
Datenbus Bit 4	ein/aus	D4	3	46	D9	ein/aus	Datenbus Bit 9
Datenbus Bit 3	ein/aus	D3	4	45	D10	ein/aus	Datenbus Bit 10
Datenbus Bit 2	ein/aus	D2	5	44	D11	ein/aus	Datenbus Bit 11
Datenbus Bit 1	ein/aus	D1	6	43	D12	ein/aus	Datenbus Bit 12
Datenbus Bit 0	ein/aus	D0	7	42	D13	ein/aus	Datenbus Bit 13
Mouse 1 Horizontal	ein	M1H	8	41	D14	ein/aus	Datenbus Bit 14
Mouse 0 Horizontal	ein	M0H	9	40	D15	ein/aus	Datenbus Bit 15
Register Address 8	ein	RGA8	10	39	M1V	ein	Mouse 1 Vertical
Register Address 7	ein	RGA7	11	38	M0V	ein	Mouse 0 Vertical
Register Address 6	ein	RGA6	12	37	GND	---	Masse
Register Address 5	ein	RGA5	13	36	CCK	ein	Color Clock
Register Address 4	ein	RGA4	14	35	7M	ein	7,15909 MHz
Register Address 3	ein	RGA3	15	34	N.C.	---	(not connected)
Register Address 2	ein	RGA2	16	33	$\overline{\text{-ZD}}$	aus	Background Indicator
Register Address 1	ein	RGA1	17	32	N.C.	---	(not connected)
Color Burst	aus	-BST	18	31	G3	aus	Video Green Bit 3
Versorgungsspannung +5V	ein	VCC	19	30	G2	aus	Video Green Bit 2
Video Red Bit 0	ein	VCC	20	29	G1	aus	Video Green Bit 1
Video Red Bit 1	aus	R1	21	28	G0	aus	Video Green Bit 0
Video Red Bit 2	aus	R2	22	27	B3	aus	Video Blue Bit 3
Video Red Bit 3	aus	R3	23	26	B2	aus	Video Blue Bit 2
Video Blue Bit 0	aus	B0	24	25	B1	aus	Video Blue Bit 1

Bild 3.2: Pinbelegung von DENISE

Funktion	Richtung	Signal	Pin	Pin	Signal	Richtung	Funktion
Datenbus Bit 8	ein/aus	D8	1	48	D9	ein/aus	Datenbus Bit 9
Datenbus Bit 7	ein/aus	D7	2	47	D10	ein/aus	Datenbus Bit 10
Datenbus Bit 6	ein/aus	D6	3	46	D11	ein/aus	Datenbus Bit 11
Datenbus Bit 5	ein/aus	D5	4	45	D12	ein/aus	Datenbus Bit 12
Datenbus Bit 4	ein/aus	D4	5	44	D13	ein/aus	Datenbus Bit 13
Datenbus Bit 3	ein/aus	D3	6	43	D14	ein/aus	Datenbus Bit 14
Datenbus Bit 2	ein/aus	D2	7	42	D15	ein/aus	Datenbus Bit 15
Masse	---	GND	7	41	RxD	ein	Serial Receive Data
Datenbus Bit 1	ein/aus	D1	9	40	TxD	aus	Serial Transmit Data
Datenbus Bit 0	ein/aus	D0	10	39	DKWE	aus	Disk Write Enable
System Reset	ein	-RES	11	38	-DKWD	aus	Disk Write Data
DMA Request Line	aus	DMAL	12	37	-DKRD	ein	Disk Read Data
Prozessor-Interrupt-Leitung	aus	-IPL0	13	36	P1Y	ein/aus	Analogport 1 Y-Spannung
Prozessor-Interrupt-Leitung	aus	-IPL1	14	35	P1X	ein/aus	Analogport 1 X-Spannung
Prozessor-Interrupt-Leitung	aus	-IPL2	15	34	DRA6	---	Analog-Masse
Interrupt-Eingang Ebene 2	ein	-INT2	16	33q	P0Y	ein/aus	Analogport 0 Y-Spannung
Interrupt-Eingang Ebene 3	ein	-INT3	17	32	P0X	ein/aus	Analogport 0 X-Spannung
Interrupt-Eingang Ebene 6	ein	-INT6	18	31	AUD A	aus	Left Audio
Register Address 6	ein	RGA1	19	30	AUD B	aus	Right Audio
Register Address 5	ein	RGA2	20	29	CCKQ	ein	Color Clock Delay
Register Address 4	ein	RGA3	21	28	CCK	ein	Color Clock
Register Address 3	ein	RGA4	22	27	VCC	ein	Versorgungsspannung +5V
Register Address 2	ein	RGA5	23	26	RGA8	ein	Register Address 8
Register Address 1	ein	RGA6	24	25	RGA7	ein	Register Address 7

Bild 3.3: Die Pinbelegung von PAULA

schiedlichen Chips zugeordnet sind. Die Bilder 3.1., 3.2 und 3.3 geben die Pinbelegungen von AGNUS, DENISE und PAULA an, das Bild 3.4 übersichtliche Blockschaltbilder mit den wichtigsten internen Funktionseinheiten. Daran lassen sich schon einige der vielseitigen in den Spezialbausteinen schlummernden Möglichkeiten erkennen.

PAULA ist für den Sound, für die Steuerung der Control Ports sowie für die Handhabung der RS-232-Datenübertragung und der Diskettenschnittstelle zuständig. Außerdem wurde hier ein programmierbarer Interrupt-Controler untergebracht, der jeder Interruptquelle des Systems eine bestimmte Unterbrechungsebene zuordnet und entscheidet, welcher Baustein den Hauptprozessor 68000 jeweils bei seiner Arbeit unterbrechen darf.

AGNUS besorgt die Speicherverwaltung des 512 Kilobyte umfassenden Chip-RAM, die Kommunikation der drei Spezial-Chips mit dem 68000-Prozessor und die Datenübermittlung per DMA (Direct Memory Access = direkter Speicherzugriff). Außerdem enthält dieser Baustein den Koprozessor (Copper) und den Blitter (Block Image Transferer). Der Copper ist dabei für die Aufbereitung der Bildsynchronsignale und damit zusammenhängende Manipulationen zuständig (genauere Hinweise zum Thema Video-Aufbereitung enthält Kapitel 4), der Blitter dagegen dient beispielsweise zum schnellen Verschieben großer Datenmengen im Speicher bzw. zum blitzartigen Füllen von Flächen.

DENISE schließlich verwaltet die Grafik, stellt Sprites dar und besorgt die Auswertung der Maus-Bewegungen. Durch die universell einsetzbare Video-Elektronik wird eine weitgehend Fernsehsystem-unabhängige Bildausgabe in verschiedenen Auflösungsstufen realisiert. Ebenso ist es möglich, die Bildausgabe auf andere Geräte zu synchronisieren und die Bilder miteinander zu mischen.

3.2 Registrierung

Zur Programmierung der drei Spezialchips und für Rückmeldungen sind ähnlich wie beim 8520 Register vorgesehen, die sich über acht gemeinsam benutzte Adreßleitungen ansprechen lassen. Damit wären maximal 256 verschiedene 16-Bit-Register möglich, von denen jedoch nicht alle belegt sind. Je nach Funktion bestehen einige Register wie gesagt aus verteilten Speicherzellen in mehreren Bausteinen. Meist gibt es unterschiedliche Adressen zum Lesen und Schreiben. Die Register werden auch von den Chips selbst benutzt, um untereinander Informationen auszutauschen. Im Adreßraum des Prozessors 68000 erscheinen die Chip-Register auf den Adressen $DFF000 bis $DFF1FF (siehe Bild 5.1 auf Seite 191).

Über die einzelnen Steuerbits lassen sich alle Systemfunktionen direkt manipulieren. Zwar verfügt das Betriebssystem des Amiga über Routinen, die dem Programmierer die lästige Bitfummelei ersparen, und in der Regel sollten auch diese Routinen benutzt werden, doch gerade bei schnellen Vorgängen ist es manchmal unumgänglich, direkt auf die Hardware zuzugreifen, um wirklich das letzte aus dem System herauszuholen.

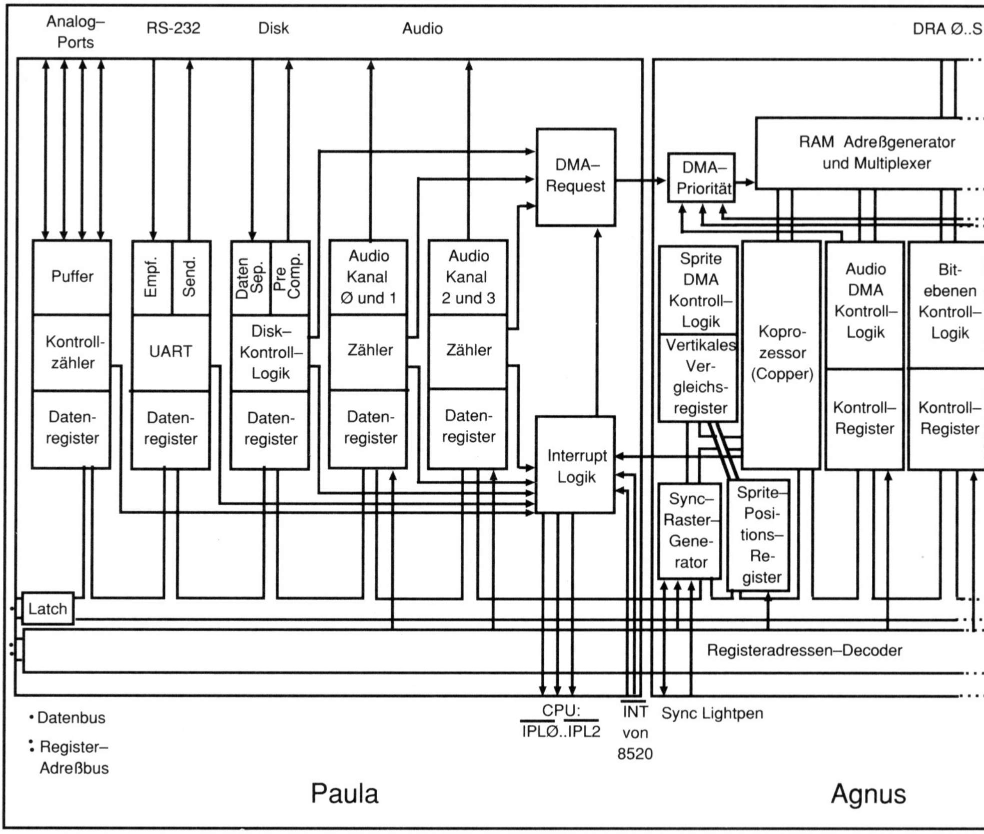
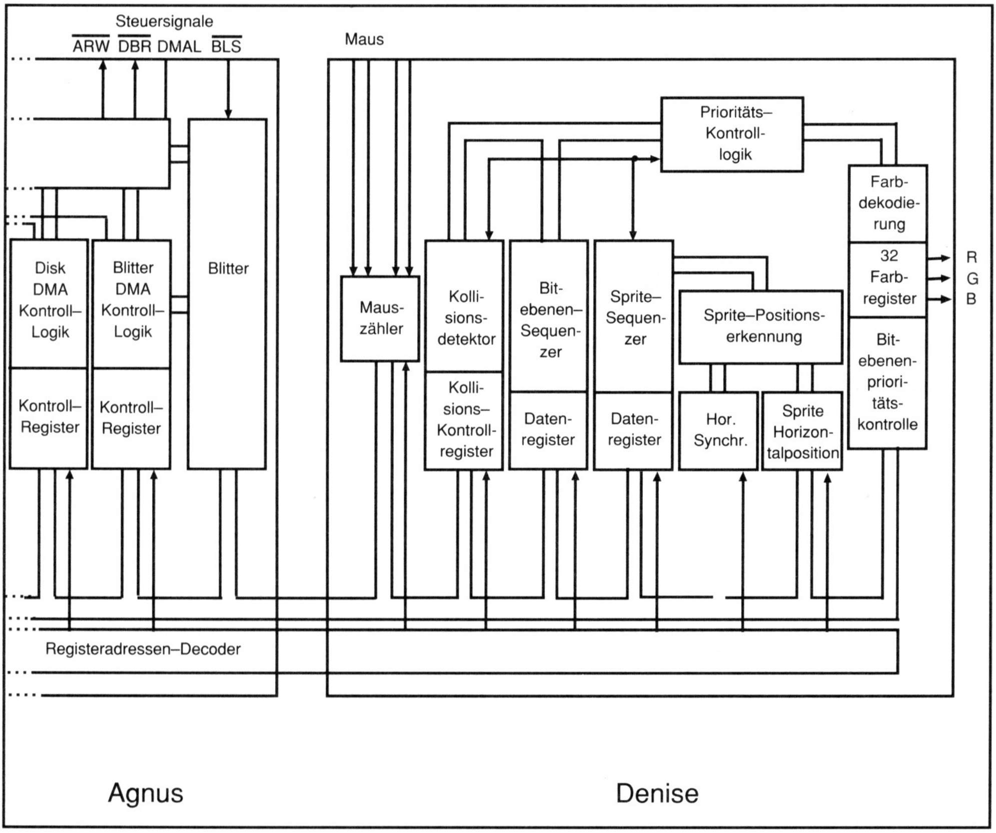


Bild 3.4: Die Kastenchips des Amiga

Im folgenden sollen einige interessante Anwendungsmöglichkeiten genauer erläutert werden. Beginnen wir mit den Control-Ports. Ihre Einsatzmöglichkeiten sind mit dem Anschluß der Maus oder eines Joysticks längst noch nicht ausgeschöpft.

3.3 Die beiden Control-Ports

Am Amiga befinden sich zwei neunpolige Min-D-Stecker, die zum Anschluß von unterschiedlichen Bedien- und Eingabegeräten vorgesehen sind. Glücklicherweise hat sich Commodore beim Amiga für die allgemein übliche neunpolige Min-D-Steckverbindung entschieden, so daß gängige und dadurch preisgünstig erhältliche Zusätze angeschlossen werden können, obwohl generell der ungeschützte Einbau der männlichen Steckerhälfte im Computer nicht die beste aller Lösungen darstellt. Die offenliegenden Pins können unbeabsichtigt berührt werden. Ist man zufällig statisch aufgeladen – was bei syntheti-



Pin	Maus	Joystick/Paddle	Lightpen
1	MOUSE V	FORWARD	–
2	MOUSE H	BACK	–
3	MOUSE VQ	LEFT	–
4	MOUSE HQ	RIGHT	–
5	BUTTON 2	POT X	LP PRESS
6	BUTTON 1	FIRE	LIGHT PEN
7	+5V	+5V	+5V
8	GND	GND	GND
9	BUTTON 3	POT Y	–

Tabelle 3.1: Die Belegung der Control-Ports

scher Kleidung und Teppichboden nicht selten vorkommt – kann das unter Umständen ganz schnell zur Zerstörung der angeschlossenen MOS-Bausteine führen!

Es ist empfehlenswert, die beiden Control-Ports mittels eines Pappstreifens abzudecken, der einseitig mit Tesafilm am Gehäuse befestigt ist, und bei Bedarf einfach hochgeklappt wird, falls nicht ohnehin ständig beide Buchsen beschaltet sind.

Tabelle 3.1 gibt die Belegung der Buchsen in den verschiedenen Betriebsarten wieder.

3.3.1 Maus & Co.

Jeder Amiga wird mit einer Maus ausgeliefert. Wie funktioniert solch ein Eingabegerät? Nun, genauso wie ein Trackball. In beiden Geräten werden je nach Bewegungsrichtung Impulse für Vertikal- und Horizontalkomponente erzeugt. DENISE besitzt dazu jeweils zwei Eingänge für eine Maus am Control-Port 0 (M0V, M0H) und am Control-Port 1 (M1V, M1H).

Um auch die Bewegungsrichtung unterscheiden zu können, sendet die Maus jedoch für beide Komponenten jeweils zwei Signale, die in einer speziellen Modulationsform kodiert sind.

Bild 3.5 gibt einen Ausschnitt aus dem Amiga-Schaltplan wieder und zeigt die weitere Verarbeitung.

Die Impulse gelangen zunächst auf einen Multiplexer, der mit dem Takt C1 abwechselnd beide Signale auf den entsprechenden Eingang von DENISE durchschaltet. Dort wird aus dem Gemisch die Bewegungsrichtung erkannt und ein 8-Bit-Zähler erhöht oder erniedrigt. Bei Bewegung nach rechts oder auf den Bediener zu wird

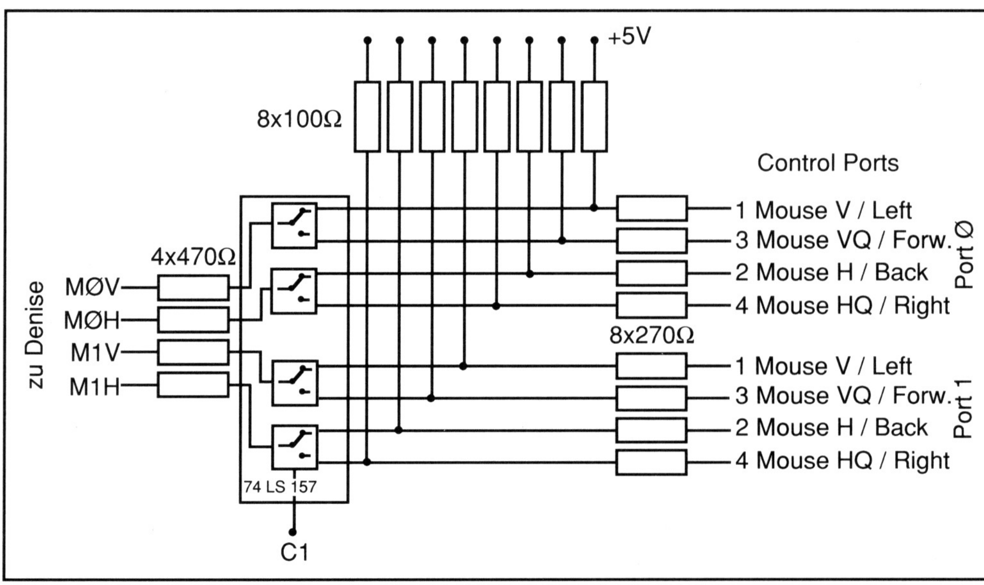


Bild 3.5: Auswertung der Maus-Signale

Register	Adresse	Bedeutung der Bits 15 ... 8	7 ... 0
JOY0DAT	$DFF00A	Vertikal	Horizontal
JOY1DAT	$DFF00C	Vertikal	Horizontal

Tabelle 3.2: Die Mauszähler

hochgezählt, bei Bewegung nach links oder vom Bediener weg dagegen abgezogen.

Die beiden Zähler für einen Control-Port sind jeweils zusammen in einem 16-Bit-Register lesbar, nämlich in JOY0DAT für Port 0 und JOY1DAT für Port 1. Die Bits 0 bis 7 sind dabei für die Horizontalrichtung zuständig, die Bits 8 bis 15 für die Vertikale. Tabelle 3.2 macht das deutlich.

Es ist empfehlenswert, die Mauszähler während jeder Vertikal-Austastlücke des Video-Bildes zu lesen. Sie enthalten dann die Bewegung seit der letzten Abfrage. Diese Werte müssen vom Programm auf den zwischengespeicherten Wert der aktuellen Position umgerechnet werden.

An die Control-Ports des Amiga können Mäuse mit bis zu drei Tasten angeschlossen werden, obwohl die Original-Amiga-Maus nur zwei Tasten besitzt. Bild 3.6 zeigt die Pinanordnung der Control-Ports, Tabelle 3.3 gibt ihre Zuordnung bei Mausbetrieb wieder.

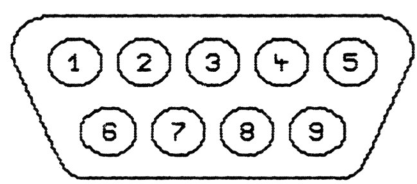


Bild 3.6: Sub-D, 9-pol, männlich, Draufsicht Steckerseite

Pin	Signal	Funktion	Anschluß
1	MOUSE V	Mauszähler vertikal	MUX (DENISE)
2	MOUSE H	Mauszähler horizontal	MUX (DENISE)
3	MOUSE VQ	Vertikale Quadratur	MUX (DENISE)
4	MOUSE HQ	Horizontale Quadratur	MUX (DENISE)
5	BUTTON 2	Maustaste 2 (n.c)	PAULA PX
6	BUTTON 1	Maustaste 1 (links)	8520-A PA7/6
7	+5V	Versorgungsspannung	+5V
8	GND	Signalmasse	GND
9	BUTTON 3	Maustaste 3 (rechts)	PAULA PY

Tabelle 3.3: Belegung der Control-Ports bei Mausbetrieb

3.3.2 Spaßig

Das erste Zusatzgerät eines Spiele-Freaks ist meist ein Joystick. Seine Funktion ist nicht schwer zu durchschauen. Er besitzt im Inneren fünf Taster (vier für den Steuerknüppel und einen für den Feuerknopf), die bei Betätigung den jeweiligen Steckeranschluß nach Masse kurzschließen. Dazu werden die gleichen Control-Port-Pins benutzt wie beim Mausbetrieb. Die vom Joystick erzeugten logischen Pegel werden daher genau wie die der Maus behandelt und die Ausgabe erfolgt auch über das Maus-Zählregister, wie die Übersicht in Tabelle 3.1 bereits zeigt. Das hat allerdings zur Folge, daß zur Ermittlung der Joystick-Positionen »vorn« und »hinten« jeweils eine XOR-Verknüpfung nötig wird. Tabelle 3.4 zeigt die Möglichkeiten.

Richtung	BIT
rechts	1
links	9
hinten	(1 XOR 0)
vorn	(9 XOR 8)

Tabelle 3.4: Ermittlung der Joystick-Positionen

3.3.3 Paddles

Ein Joystick erlaubt nur sehr grobe Eingaben: links, rechts, oben, unten sowie Feuerknopf gedrückt oder nicht. Paddles dagegen sind Regler, die stufenlos verstellbar sind. Sie werden daher auch analoge Eingabegeräte genannt, während Joysticks immer digital arbeiten. Ein Paddle-Regler im Selbstbau und sein Einfluß auf das System soll nun beschrieben werden.

3.3.3.1 Drehen statt Drücken

Das Funktionsprinzip von Paddles ist einfach und schnell erklärt: In PAULA sitzen vier Analog/Digital-Wandler, die zu einem Widerstandswert am jeweiligen Eingang eine Bitkombination erzeugen, und diese in einem Register festhalten. Der Anschluß am Control-Port wurde bereits in der Übersicht aus Tabelle 3.1 erwähnt und erfolgt intern nach Bild 3.7.

Zur Bestimmung der Reglerstellung wird ein Kondensator über den eingestellten Widerstand aufgeladen. Das dauert je nach Widerstandswert eine gewisse Zeit. Während der vertikalen Austastlücke des Videosignals muß der Wandelvorgang gestartet werden. Das geschieht durch Beschreiben des Registers POTGO (Adresse

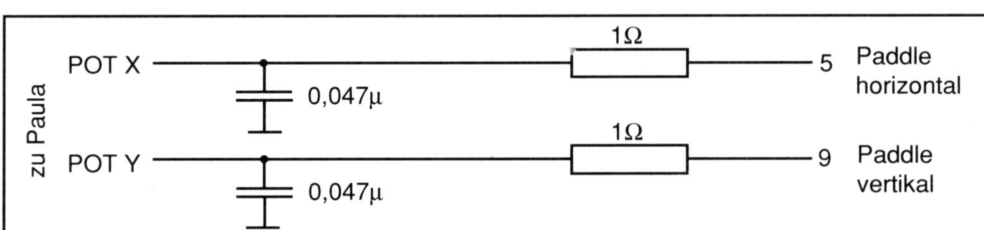


Bild 3.7: Auswertung der Paddle-Spannungen

Register	Adresse	Bedeutung der Bits 15 8	7 0
POT0DAT	$DFF012	POT0 Y-Wert	POT0 X-Wert
POT1DAT	$DFF014	POT1 Y-Wert	POT1 X-Wert

Tabelle 3.5: Die A/D-Wandler-Register

$DFF034) mit dem Wert $0001. Daraufhin wird der interne Kondensator entladen und abgewartet, bis die horizontale Video-Position Zeile 7 oder 8 erreicht hat. Dann beginnt die Ladung des Kondensators, wobei seine Spannung hardwaremäßig ständig mit einem festen Wert verglichen wird. Zu Anfang jeder neuen Video-Zeile wird nun der Ausgabewert erhöht, falls die feste Spannung noch nicht erreicht war. Ansonsten erfolgt keine Änderung mehr. Der Wert bleibt bestehen, bis zum nächsten Schreibvorgang auf POTGO.

Diese Meßmethode erlaubt recht genaue Ergebnisse und kostet keine CPU-Zeit, da alle Vorgänge rein hardwaremäßig ablaufen. Allerdings ist sie recht langsam. Tabelle 3.5 zeigt die Ausgaberegister und die Verwendung der einzelnen Bits.

3.3.3.2 Fast rein mechanisch

Ein Handregler, im Computerjargon Paddle genannt, läßt sich leicht selber bauen.

Bereits ein regelbarer Widerstand, vom Analogeingang nach +5V geschaltet, liefert über den Wandler genau einstellbare Abtastwerte. Vorgeschlagen wird ein Poti mit 470 Kiloohm ± 10%. Es ist jedoch gut möglich, daß am Anfang oder am Ende des Regelbereiches keine Widerstandsänderungen mehr registriert werden, weil die Meßeinrichtung bereits ihren Minimal- bzw. Maximalwert erreicht hat. Um diese Fälle zu vermeiden, sind zusätzlich zwei Trimmpotentiometer vorzusehen, über die sich der Widerstandsbereich genau den Erfordernissen anpassen läßt.

Bild 3.8 zeigt die entsprechende Schaltung. Sie enthält außerdem einen Feuerknopf, der bei Paddles nicht der Verdrahtung in Joysticks entspricht!

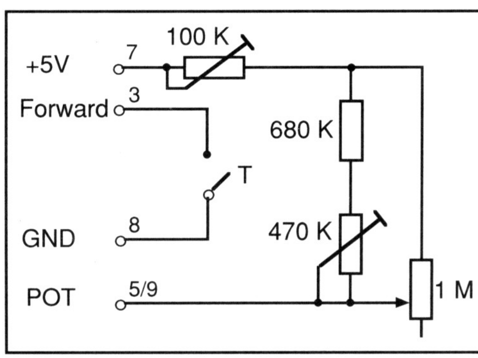
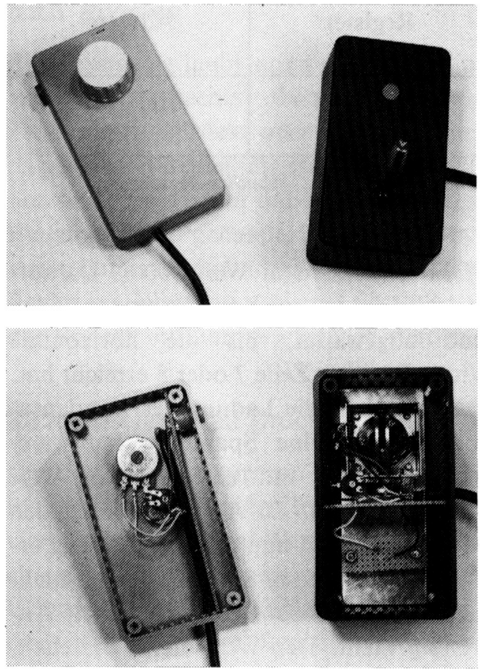


Bild 3.8: Der optimal abgleichbare Paddle-Regler

Zum Einbau des Reglers wird ein handliches und formschönes Gehäuse mit den ungefähren Abmessungen 160 x 60 x 20 mm ausgewählt, das an den Innenflächen der Seitenwände Stege zum Einstecken von Platinen aufweist.

Zuerst werden drei Bohrungen angebracht, und zwar für das Drehpotentiometer auf der Frontseite, für den Feuerknopf an der linken (bzw. für Linkshänder rechten) Seitenwand oben, und für die Durchführung der Zuleitung an der unteren Stirnseite.

Ein schmaler Streifen wird nun aus einer Lochrasterplatte gebrochen, der Feuerknopf darauf gesteckt und die Anordnung in das Gehäuse eingepaßt. Es ist dabei empfehlenswert, einen runden Knopf zu verwenden, um die benötigte Öffnung im Gehäuse leicht herstellen zu können. Auch sollte der Taster einen deutlichen Druckpunkt besitzen; etwa wie der Feuerknopf eines Joysticks auch. Als Drehpotentiometer eignet sich am besten eine Ausführung mit Metallachse, die besonders angenehme Bedienungseigenschaften aufweist. Diese Achse wird auf die benötigte Länge (je nach Knopf) abgesägt und der Regler festgeschraubt.



Fot 3.1: Eindimensionale und zweidimensionale Paddles

Die genügend lang abisolierte Zuleitung kann mittels zweier blanker Schaltdrähtchen an der Lochrasterplatte festgeklemmt werden.

Die Anordnung wird nun noch nach dem Schaltplan (Bild 3.8) verlötet, wobei die Trimmpotentiometer freitragend einzufügen sind. Es dürfen auch ruhig beide Eingänge für X- und Y-Wert am Schleifer angelötet werden.

3.3.3.3 Optimaler Abgleich

Bevor der Deckel aufgeschraubt wird, sollten Sie noch die Abgleicharbeiten durchführen. Benutzen Sie dazu das Programm POTTest, das ständig den aktuellen Meßwert einliest und auf dem Bildschirm darstellt.

Der Drehregler wird ganz an den linken Anschlag gedreht. Der Wert des Trimmpotentiometers P1 sollte nun von oben her so verstellt werden, daß auf dem Bildschirm gerade noch 0 angezeigt wird. Anschließend wird der Regler in die andere Extremstellung gebracht und P2 von unten her so verstellt, daß gerade 255 erscheint. Damit ist der Paddle-Zusatz optimal abgeglichen und nach Verschrauben des Deckels einsatzbereit.

In der gleichen Weise kann auch ein X-Y-Paddle hergestellt werden, indem man je-

Bit	Name	Funktion	Anschluß
15	OUTRY	Output Enable für Bit 14	
14	DATRY	I/O Data	Port 1, Pin 9
13	OUTRX	Output Enable für Bit 12	
12	DATRX	I/O Data	Port 1, Pin 5
11	OUTLY	Output Enable für Bit 10	
10	DATLY	I/O Data	Port 0, Pin 9
9	OUTLX	Output Enable für Bit 8	
8	DATLX	I/O Data	Port 0, Pin 5
7–1	X	(reserviert für Chip Identifikation)	
0	START	POT-Messung starten	
Lesen ausschließlich unter POTINP ($DFF016), Schreiben nur unter POTGO ($DFF034)			

Tabelle 3.6: I/O-Kontrolle der POT-Pins

des der beiden Potis eines Kreuzknüppels nach Bild 3.8 verschaltet, und einen Ausgang an Pin 5 des Controlport-Steckers, den anderen an Pin 9 anschließt.

3.3.4 Noch mehr Ports

Das war immer noch nicht alles, was mit den Control-Ports anzufangen ist. Ein weiteres Register dient nämlich als Portregister und erlaubt die Umschaltung der Analogeingänge zu universell verwendbaren I/O-Ports. Die Bedeutung der Registerbits zeigt Tabelle 3.6.

Bitte beachten Sie, daß dieses Register für Lesen und Schreiben unterschiedliche Adressen besitzt. Wird eines der Output-Enable-Bits gesetzt, so geht der entsprechende POT-Anschluß am Control-Port in

Pin	Signal	Funktion	Anschluß
1	FORWARD	Joystick nach oben	DENISE
2	BACK	Joystick nach unten	DENISE
3	LEFT	Joystick nach links	DENISE
4	RIGHT	Joystick nach rechts	DENISE
5	POTX	Paddle horizontal	PAULA PX
6	FIRE	Feuerknopf	8520-A PA7/6
7	+5V	Versorgungsspannung	+5V
8	GND	Signalmasse	GND
9	POTY	Paddle vertikal	PAULA PY

Tabelle 3.7: Belegung der Control-Ports bei Paddlebetrieb

den Ausgabe-Modus und der augenblickliche Zustand des zugehörigen Datenbits erscheint am Control-Port. Gleichzeitig wird der A/D-Wandler vom betroffenen Kanal abgetrennt. Er kann nun nicht mehr benutzt werden.

Aufgrund der relativ großen Kapazität der zwischen die Leitung zu PAULA und Masse geschalteten Ladekondensatoren (0,047 Mikrofarad) muß beim Beschreiben der Port-Ausgänge leider mit einer recht langen Reaktionszeit von bis zu 300 Mikrosekunden gerechnet werden. Tabelle 3.7 zeigt die Pinbelegung der Control-Ports bei Paddle- bzw. Port-Betrieb.

3.4 Die RS-232-Schnittstelle des Amiga

Schauen wir uns nun die RS-232-Schnittstelle etwas genauer an, die nicht mit den seriellen Ports in den I/O-Bausteinen zu verwechseln ist. Vorab einige allgemeine Informationen. Dann einige Geräte, die diese Schnittstelle nutzen, z.B. ein Modem und eine Midi-Schnittstelle.

3.4.1 Grundlagen

Die Grundidee bei der seriellen Verbindungsart ist, die Daten nicht byteweise, sondern Bit für Bit als Spannungsimpulse zu übermitteln. Dies geschieht in einer vorher genau festgelegten Geschwindigkeit, damit der Empfänger den Sender überhaupt verstehen kann.

3.4.1.1 Nicht zu bremsen

Die Übermittlungsgeschwindigkeit wird in Baud gemessen. Ein Baud ist bei digitaler Übertragung genau ein Bit pro Sekunde. Tabelle 3.8 gibt die gebräuchlichen Baudraten an.

50 Bit/s	1200 Bit/s
110 Bit/s	2400 Bit/s
150 Bit/s	4800 Bit/s
300 Bit/s	9600 Bit/s
600 Bit/s	19200 Bit/s

Tabelle 3.8: Übliche Baudraten bei serieller Übertragung

Um Sender und Empfänger zeitlich exakt aufeinander abstimmen, also synchronisieren zu können, werden in der Praxis Pakete von 5 bis 8 Datenbits übertragen, die von Start- und Stopbits eingerahmt sind. Das Startbit hat grundsätzlich LOW- und das Stopbit HIGH-Pegel. Dadurch entsteht bei der Übertragung zwischen zwei Byte immer eine fallende Flanke, aus der eine geeignete Vorrichtung im Empfänger (Soft- oder Hardware) den Sendetakt zurückgewinnen kann. Eine Datenübertragung mit einer einzigen zweiadrigen Leitung ist damit durchaus möglich.

Vor dem Stopbit kann zur Fehlererkennung noch ein sogenanntes Paritäts-Bit vereinbart werden, das die Anzahl der HIGH-Zustände im Datenpaket immer gerade oder ungerade macht. Sind zum Beispiel in einer 8-Bit-Übertragung 5 Bits HIGH, wird bei gerader Parität das Paritäts-Bit vom Sender ebenfalls gesetzt. Der Empfänger überprüft nun den Zustand der Bits und ist damit in der Lage, Einzelfehler zu erkennen.

Um die Störanfälligkeit besonders bei Langstreckenübertragungen zu vermin-

dern, wurde bei der RS-232-Norm von den sonst üblichen TTL-Pegeln abgewichen. Schon der Leitungswiderstand des Kabels würde auf sehr langen Strecken nämlich bewirken, daß ausgesendete HIGH-Pegel beim Empfänger nicht mehr mit 5 Volt ankämen, sondern vielleicht nur noch mit 2 Volt, was bereits als ein LOW-Signal interpretiert werden müßte. Man benutzt Spannungen von –12 bis –3 Volt für ein gesetztes Bit und +3 bis +12 Volt für eine logische Null. Wegen der sich um maximal 24 Volt unterscheidenden Potentiale wurde die RS-232-Übertragung hierzulande V.24-Norm getauft (DIN 66020). Allein die Definition, welche Spannung für 1 und welche für 0 steht, ist unterschiedlich, was allgemein zu einiger Verwirrung führt. Hier kann nur der Rat gegeben werden: Nicht auf Beipackinformationen verlassen, sondern im Einzelfall selbst nachprüfen! Auch bei den Herstellern wird oft manches durcheinandergeworfen. Im Falle eines Falles helfen zwischengeschaltete Negierer weiter.

3.4.1.2 Gegenverkehr

Wie gesagt ist die Übertragung von Daten grundsätzlich mit den dargestellten Vereinbarungen über eine nur zweiadrige Verbindung möglich. Die Bezeichnung der Sendeleitung ist allgemein TXD (Transmit Data = Sendedaten); die andere Ader stellt natürlich die Masseverbindung her.

Will man gleichzeitig auch Daten wieder zurückschicken, wird eine dritte Ader nötig. Ihre Bezeichnung ist gewöhnlich RXD (Receive Data = Empfangsdaten). Damit können auch Geräte bedient werden, die nicht so schnell arbeiten wie der Sender, indem auf der Rückleitung ein bestimmtes Zeichen übermittelt wird, um den Sender erst einmal zu stoppen. Es handelt sich um ein sogenanntes Softwareprotokoll, wovon das gebräuchlichste das XON-XOFF-Protokoll ist. Dabei werden die beiden ASCII-Steuerzeichen DC3 (XOFF = $13 = CTRL-S) und DC1 (XON = $11 = CTRL-Q) verwendet. Mit XOFF stoppt der Datenempfänger die Sendung seiner Gegenstation, während er intern zum Beispiel mit Ausdrucken oder Abspeichern beschäftigt ist, mit XON dagegen meldet er seine Bereitschaft zur Aufnahme weiterer Zeichen.

Sender und Empfänger müssen dann allerdings in der Lage sein, gleichzeitig zu senden und zu empfangen.

Es geht auch anders, wie die ETX-ACK-Prozedur zeigt. Der Computer sendet hier einen ganzen Datenblock, der mit ETX (End Of Text = Textende = $03) abgeschlossen wird. Sobald er von der Gegenstation komplett verarbeitet ist, sendet diese das Steuerzeichen ACK (Acknowledge = Bestätigung = $06) zurück, woran der Computer erkennt, daß er jetzt den nächsten Datenblock senden kann.

Die ETX-ACK-Prozedur setzt mindestens empfangsseitig einen Pufferspeicher voraus, da der Empfänger ja die Übertragung während eines Datenblocks nicht anhalten kann.

Falls es die Hardware erlaubt, ist es meist doch wieder einfacher, eine zusätzliche Handshakeleitung zu realisieren: DTR

Signalname	Richtung	Bedeutung
TXD	Ausgang	Transmitted Data Sendedaten vom Rechner
RXD	Eingang	Received Data Empfangsdaten vom Peripheriebaustein
DSR	Eingang	Data Set Ready Peripheriebaustein betriebsbereit
CTS	Eingang	Clear To Send Peripheriebaustein empfangsbereit
DTR	Ausgang	Data Terminal Ready Zeigt dem Peripheriegerät an, daß die serielle Schnittstelle des Amiga bereit ist (entspricht DSR)
RTS	Ausgang	Request To Send Zeigt dem Peripheriegerät an, daß der Amiga über seine serielle Schnittstelle Daten senden will (entspricht CTS)
CD	Eingang	Carrier Detect Modemsignal, das den Empfang einer Gegenstelle anzeigt
RI	Eingang	Ring Indicator Modemsignal, das einen Anruf anzeigt

Tabelle 3.9: Bedeutung der RS-232-Kontrollsignale

(Data Terminal Ready = Datenempfänger bereit). Kann der Empfänger keine Daten mehr aufnehmen, braucht er nämlich lediglich diese Leitung zu desaktivieren. Der Sender wartet daraufhin so lange, bis das Signal wieder erscheint. Tabelle 3.9 zeigt die Bedeutung aller Steuerleitungen der seriellen Schnittstelle am Amiga.

3.4.1.3 Rahmenbedingungen

Die serielle Schnittstelle des Amiga wird durch drei Register gesteuert. Tabelle 3.10 enthält ihre wichtigsten Daten.

Zunächst ist die Übertragungsgeschwindigkeit einzustellen. Dazu sind die Bits 14 bis 0 des Parameterregisters SERPER (Serial Port Period and Control, Adresse $DFF032) vorgesehen. Dem Übertragungstakt liegt die Farbtaktfrequenz Color Clock (CCK, etwa 3,5 MHz) zugrunde, die durch den um 1 erhöhten eingestellten Wert geteilt wird. Falls N die mit den genannten Bits festgelegte Zahl ist, beträgt der Übertragungstakt

$$\frac{CCK}{(N + 1)}$$

Register	Adresse	Bedeutung	Funktion
SERPER	$DFF032	Serial Port Period and Control Register	Parameter- und Wortlänge
SERDATR	$DFF018	Serial Port Status and Data Read	Empfangspuffer u. Statusregister
SERDAT	$DFF030	Serial Port Data and Stop Bits Write	Sendepuffer und -Parameter

Tabelle 3.10: Die Register zur Steuerung der RS-232-Übertragung

bzw. die Baudrate entsprechend ein Bit pro

$$(N + 1) * \frac{1}{CCK}$$

Bit 15 des selben Registers legt fest, ob es sich um eine Übertragung mit 8 oder 9 Datenbits handeln soll. Bit 15=0 entspricht 8 Datenbits, Bit 15=1 dagegen 9 Datenbits.

Nun folgen die nötigen Datenregister, die jedoch selbst nur Pufferspeicher bilden. Der eigentliche Datenaustausch geschieht über Schieberegister, die vom Programmierer her unzugänglich sind. Bei jeder Übertragungstaktphase wird der Zustand des Eingangs in das Schieberegister übernommen. Dabei wird der bisherige Inhalt dieses Registers nach links geschoben und das neue Bit rechts, also an der niederwertigsten Stelle, angefügt. Ist dies achtmal geschehen, dann wird der empfangene Wert selbsttätig vom Schieberegister in das Empfangspufferregister übertragen und gleichzeitig RBF (Bit 14) im Empfangsregister SERDATR, wie auch im Interrupt-Request-Register (Bit 11) gesetzt, um anzuzeigen, daß ein neues Datenwort bereitsteht. Das Schieberegister ist damit bereit zum Empfang des nächsten Wortes.

Während der Datenempfang läuft, kann auf dem anderen Kanal davon unabhängig gesendet werden.

Schreibt man einen Wert in den Sendepuffer, wird er in das Schieberegister übernommen, sobald dieses leer ist. Gleichzeitig wird – analog zum Empfang von Daten – TBE (Bit 13) im Empfangsregister SERDATR sowie das entsprechende Bit im Interrupt-Request-Register gesetzt, um anzuzeigen, daß der Baustein ein weiteres Wort im Senderegister SERDAT erwartet.

Der Inhalt des Schieberegisters wird dann bei jedem Übertragungstakt Bit für Bit auf den seriellen Ausgang gelegt. Dabei ist auch hier die Schieberichtung wieder links, so daß das höchstwertige Bit zuerst erscheint. Sind alle Bits herausgeschoben, wird das nächste Wort aus dem Sendepuffer übernommen.

Falls trotz des Interrupts kein neues Byte ins Datenregister geschrieben wurde, geht TSRE (Bit 12) des Empfangsregisters auf 1 und die serielle Datenausgabe hält an.

Bit	Name	Bedeutung (falls gesetzt)
15	OVRUN	Overrun-Bit Erscheint auch im Interrupt-Request-Register Zeigt an, daß ein empfangenes Byte vor dem Auslesen bereits von einem anderen überschrieben wurde. (Abhilfe: RBF im Interrupt-Request-Register rücksetzen)
14	RBF	Read Buffer Full Erscheint auch im Interrupt-Request-Register Datenwort empfangen. Nach dem Auslesen muß RBF im Interrupt-Request-Register gelöscht werden.
13	TBE	Transmit Buffer Empty Sendepuffer leer. SERDAT erwartet das nächste Datenwort. (Achtung: interrupt nur, wenn dieses Register gerade leer wird) (Benutzt für Vollduplex-Übertragung)
12	TSRE	Transmit Shift Register Empty Sende-Schieberegister leer. Es wurde kein Datenwort mehr ins Output-Register (SERDAT) geschrieben. Lückenlose Übertragung beendet (Automatische Löschung bei neuen Daten) (Benutzt für Halbduplex-Übertragung)
11	RXD	PAULA RxD Pin Original-Empfangsdaten
10		(unbenutzt)
9	STP	Stop Bit Stopbit, falls für den Empfänger 9 Datenbits eingestellt sind (Register SERPER)
8	STP DB8	Stop Bit Stopbit, falls für den Empfänger 8 Datenbits eingestellt sind (Register SERPER) o d e r 9. Datenbit, falls für den Empfänger 9 Datenbits eingestellt sind (Register SERPER)
7	DB7	8. Datenbit
6	DB7	7. Datenbit
5	DB7	6. Datenbit
4	DB7	5. Datenbit

Bit	Name	Bedeutung (falls gesetzt)
3	DB7	4. Datenbit
2	DB7	3. Datenbit
1	DB7	2. Datenbit
0	DB7	1. Datenbit
Register lesbar unter Adresse $DFF018		

Tabelle 3.11: Bedeutung der Bits im Empfangsregister SERDATR

3.4.1.4 Unterbrechung nötig

Wie Sie sehen, enthält das Empfangsregister nicht nur das empfangene Datenwort, sondern auch spezielle Statusinformationen. Tabelle 3.11 führt sie im einzelnen auf.

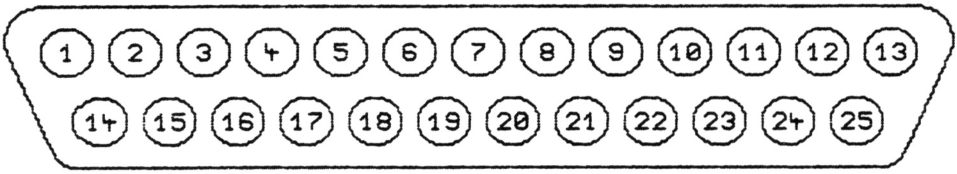
Zu sendende Daten werden linksbündig in das serielle Ausgabedatenregister SERDAT geschrieben, das heißt, Bit 0 bis Bit 7 bzw. Bit 8 enthalten die Daten. Das nächsthöhere Bit stellt das Stopbit dar und muß gesetzt sein. Falls nötig, können auch mehrere Stopbits eingefügt werden. Alle anderen Bits werden gelöscht. Die Position des höchstwertigen gesetzten Bits legt die Länge des Datenwortes fest. Ein Startbit wird von der Schaltung automatisch erzeugt.

Nach dem Beschreiben des Sendepuffers beginnt die Übertragung sofort und erfolgt ebenfalls mit der im Register SERPER festgelegten Geschwindigkeit.

3.4.2 Die serielle Schnittstelle des Amiga

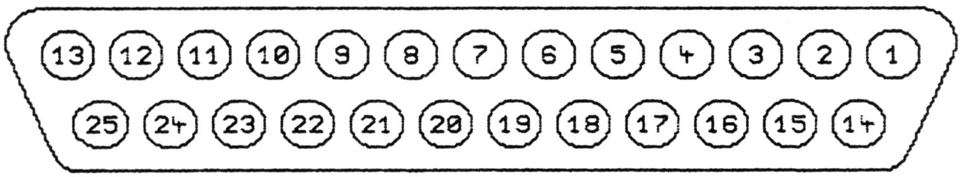
Die Amiga-Computer verfügen über eine RS-232-Normschnittstelle. Leider gelten auch hier einige der bereits in Kapitel 2.1 gemachten Aussagen, was die Kompatibilität und die Spannungsversorgung angeht. Die Pinbelegung wurde bei den neuen Modellen gegenüber dem Amiga 1000 leicht geändert. Vor allem hat man den verwendeten 25-poligen Sub-D-Stecker gegen eine komplementäre Ausführung getauscht. Die genaue Pinbelegung entnehmen Sie bitte den Tabellen 3.12 und 3.13. Dort ist auch verzeichnet, wo die Pins intern angeschlossen sind. Meist führt die Verbindung allerdings über RS-232-Pegelwandler. Näheres dazu enthält Tabelle 1.1

Wie beim parallelen Anschluß liegt auch am Steckverbinder der seriellen Schnittstelle Betriebsspannung an. Beim Amiga 1000 sind das gleich drei Spannungen, nämlich +12 Volt, +5 Volt und –5 Volt, bei allen anderen Amigas nur +12 Volt und –12 Volt. Vor allem die negativen Spannungen kommen gerade bei RS-232-Schaltungen sehr gelegen. Allerdings hat Commodore auch hier Schutzwiderstände vorgesehen, die mit 470 Ohm relativ hoch dimensioniert sind. Sollen Schaltungen direkt über die Steckverbindung versorgt werden, dann ist es



Pin	Signal	Funktion	intern	Richtg.
1	SHIELD	Schirmung	GND	
2	TXD	Sendedaten	PAULA	aus
3	RXD	Empfangsdaten	PAULA	ein
4	RTS	Sendebereitschaft	8520-A PA6	aus
5	CTS	Sendefreigabe	8520-A PA4	ein
6	DSR	Empfänger bereit	8520-A PA3	ein
7	GND	Signalmasse	GND	
8	CD	Trägererkennung	8520-A PA5	ein
9	+12V	Versorgungsspannung	+12V (470 Ohm)	
10	-12V	Versorgungsspannung	-12V (470 Ohm)	
11	AUDO	Tonausgang	linker Kanal	aus
12	---	(nicht belegt)		
13	---	(nicht belegt)		
14	---	(nicht belegt)		
15	---	(nicht belegt)		
16	---	(nicht belegt)		
17	---	(nicht belegt)		
18	AUDI	Toneingang	wird zum rechten Kanal gemischt	
19	---	(nicht belegt)		
20	DTR	Endgerät bereit	8520-A PA7	aus
21	---	(nicht belegt)		
22	RI	Rufsignalerkennung		ein
23	---	(nicht belegt)		
24	---	(nicht belegt)		
25	---	(nicht belegt)		

Tabelle 3.12: Anschlußbelegung des seriellen Ports beim Amiga 500 und 2000 (männlicher Stecker im Gerät)



Pin	Signal	Funktion	intern	Richtg.
1	SHIELD	Schirmung	GND	
2	TXD	Sendedaten	PAULA	aus
3	RXD	Empfangsdaten	PAULA	ein
4	RTS	Sendebereitschaft	8520-A PA6	aus
5	CTS	Sendefreigabe	8520-A PA4	ein
6	DSR	Empfänger bereit	8520-A PA3	ein
7	GND	Signalmasse	GND	
8	CD	Trägererkennung	8520-A PA5	ein
9	---	(nicht belegt)		
10	---	(nicht belegt)		
11	---	(nicht belegt)		
12	---	(nicht belegt)		
13	---	(nicht belegt)		
14	–5V	Versorgungsspannung	–5V	
15	AUDO	Tonausgang	linker Kanal	aus
16	AUDI	Toneingang	wird zum rechten Kanal gemischt	
17	EB	Port-Takt	GND	
18	$\overline{\text{INT2}}$	Interrupt Level 2	PAULA	ein
19	---	(nicht belegt)		
20	DTR	Endgerät bereit	8520-A PA7	aus
21	+5V	Versorgungsspannung	+5V	
22	---	(nicht belegt)		
23	+12V	Versorgungsspannung	+12V	
24	$\overline{\text{C2}}$	3,58 MHz-Takt	1/2 Proz.takt	aus
25	$\overline{\text{RESB}}$	System-Reset	gepuffert	aus

Tabelle 3.13: Anschlußbelegung des seriellen Ports beim Amiga 1000 (weiblicher Stecker im Gerät)

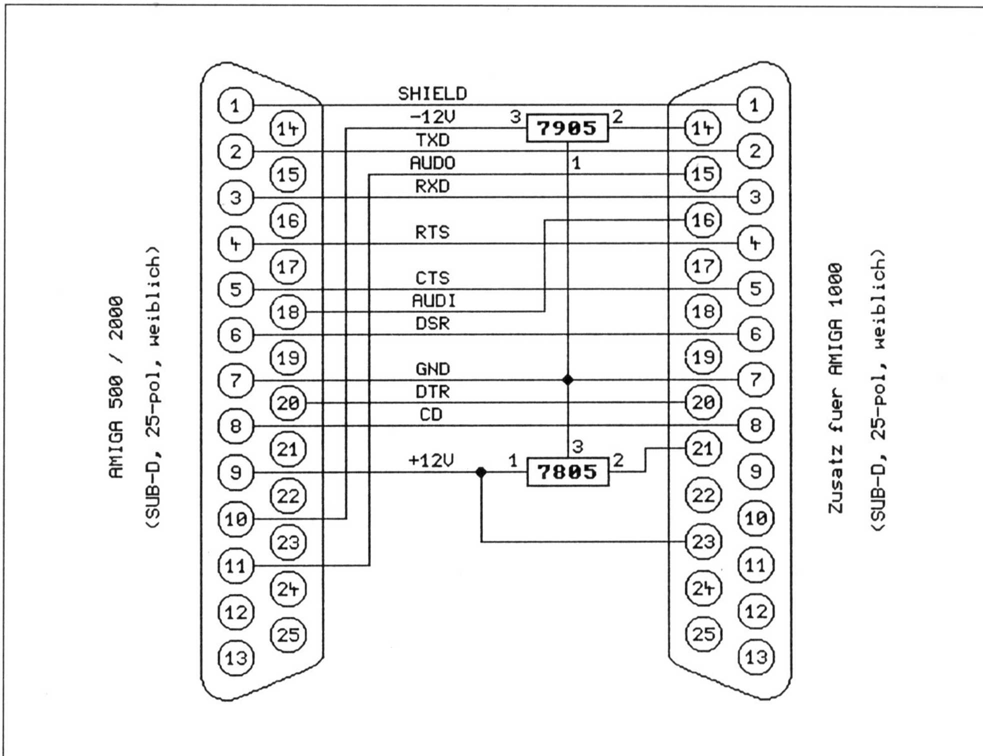


Bild 3.9: Adapter zum Anschluß eines Zusatzes für den Amiga 1000 an den RS-232-Stecker des Amiga 500 bzw. 2000

ratsam, die internen Widerstände zu überbrücken. Lesen Sie zu diesem Problem bitte Kapitel 2.1.2.

Um Geräte für den Amiga 1000 auch an den anderen Amigas betreiben zu können, benötigt man einen Adapter. Bild 3.9 zeigt eine geeignete Schaltung. Die benötigten Spannungen werden auf einfache Weise über Spannungsregler aus den vorhandenen höheren Werten gewonnen.

Ebenso läßt sich ein Gerät für den Amiga 500 oder 2000 über den Adapter nach Bild 3.10 auch an den Amiga 1000 anschließen. Der Spannungswandler ICL 7660 setzt dabei die positive +12-Volt-Betriebsspannung in ihr negatives Komplement um. Er wird bei den Platinen aus diesem Buch nicht benötigt.

3.4.3 Platz für MAX

Datenübertragung zu anderen Computern ist über die genormte RS-232-Schnittstelle kein Problem, sofern das Gegenüber ebenfalls eine solche Schnittstelle besitzt. Leider existieren aber auch Geräte, die zwar über eine serielle Schnittstelle verfü-

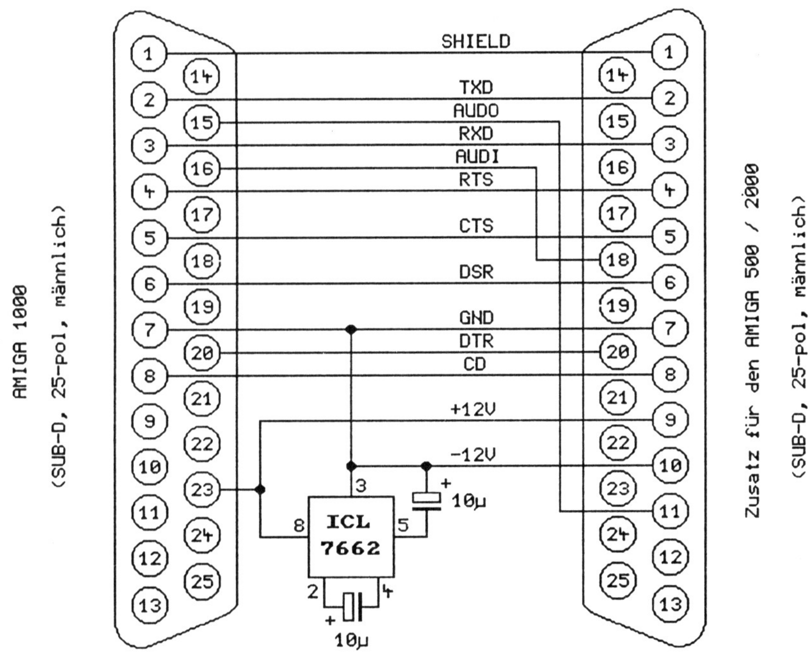


Bild 3.10: Adapter zum Anschluß eines Zusatzes für den Amiga 500/2000 an die RS-232-Schnittstelle des Amiga 1000

gen, dort jedoch mit normalen TTL-Pegel arbeiten. Als Beispiel sei hier der C64 genannt. Der direkte Anschluß einer Standardschnittstelle könnte für die Elektronik dort wegen der negativen Signalpegel tödlich ausgehen.

Glücklicherweise werden inzwischen integrierte Bausteine angeboten, die alle Signale ohne Probleme normgerecht umformen. Üblicherweise benötigen sie dazu eine symmetrische Versorgungsspannung mit +12 Volt und –12 Volt. Eine Ausnahme bildet der MAX 232, der sich mit einer einfachen Versorgungsspannung von +5 Volt begnügt.

Bild 3.11 zeigt ein Blockdiagramm zu seiner Funktion. Im oberen Teil ist ein Spannungswandler angedeutet, der die benötigten Hilfsspannungen über vier externe Elkos aus +5 Volt erzeugt. Im unteren Teil sind vier invertierende Treiber zu erkennen, von denen zwei TTL-Ein- und RS-232-Ausgänge haben, während die anderen beiden umgekehrt eine Umsetzung der RS-232-Pegel nach TTL-Norm vornehmen.

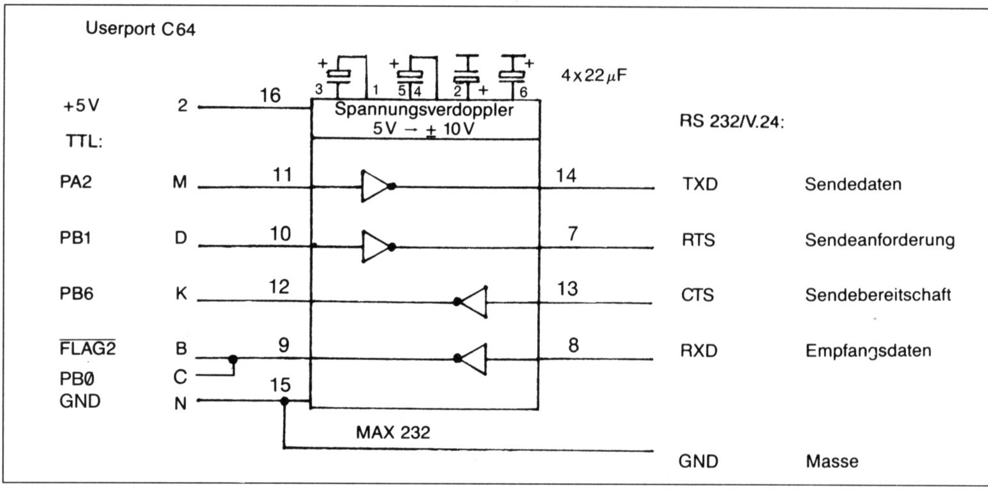


Bild 3.11: Schaltung der RS-232-Platine und Blockschaltbild des MAX 232

Damit läßt sich leicht eine normgerechte Schnittstelle nachrüsten. In Bild 3.11 sind sogar schon die entsprechenden Anschlüsse des C64-Userports sowie die Pins am RS-232-Stecker für eine bidirektionale Übertragung mit Handshakesignalen angegeben.

Bild 3.12 zeigt das Layout einer kleinen Platine für den Userport des C64, die sogar noch in einem gewöhnlichen Steckergehäuse Platz findet.

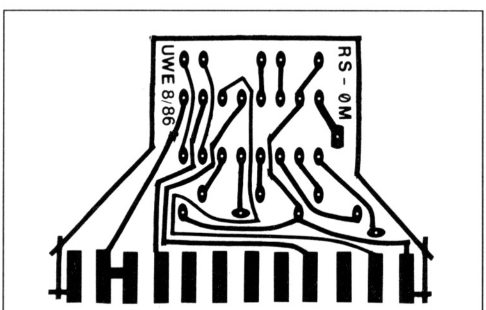


Bild 3.12: Eine Wandlerplatine für normgerechte RS-232-Signale

Den Bestückungsplan enthält Bild 3.13 und die benötigten Bauteile Tabelle 3.14.

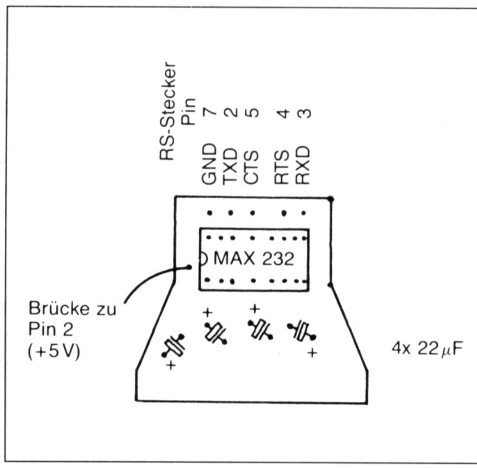


Bild 3.13: Bestückung der RS-232-Platine

Bitte richten Sie sich beim Aufbau wieder nach den Hinweisen im Anhang A. Bei dieser Platine darf jedoch das IC wegen der mangelnden Einbauhöhe nicht gesockelt werden. Aus dem selben Grund ist

1	Pegelwandler MAX 232
4	Elkos 22 Mikrofarad / 16 Volt, radial, kleine Bauform bzw. Tantal
1	Userportstecker
1	Userport-Steckergehäuse
1	einseitige Platine nach Bild 3.12

Tabelle 3.14: Bauteile für die Pegelwandler-Platine

bei den vier Elkos auf möglichst kleine Bauform zu achten. Natürlich können auch Tantal-Ausführungen verwendet werden. Eventuell müssen sie liegend in die Platine eingesetzt werden. Achten Sie besonders auch auf die richtige Polarität.

Der Userportstecker wird mit seiner unteren Kontaktleiste auf die vorgesehenen Kupferfahnen der Platine aufgelötet. Vergessen Sie nicht die Verbindung zu Pin 2 (+5V). Beim Einbau in das Gehäuse muß das RS-232-Kabel mit dem Zugentlastungsbügel gut angeschraubt werden, um Beschädigungen durch ausgerissene oder gar sich berührende Kabelenden zu vermeiden.

3.4.4 Daten um die Welt

Der wohl bekannteste Einsatzfall einer seriellen Datenübertragung ist die Benutzung des öffentlichen Telefonnetzes. Unzählige Fernsprecher auf der ganzen Welt lassen sich damit direkt anwählen und bei vielen meldet sich ein Computer.

3.4.4.1 Zweiton-Musik

Nun darf aber nicht einfach die Telefonstrippe abgeschnitten werden, um sie an den Amiga anzuklemmen, denn zum einen ist das untersagt, und zum anderen würde eine solche Übertragung auch gar nicht funktionieren. Telefonleitungen sind zur Übertragung von Tonfrequenzen gebaut – für den Wechselstrom also, der durch die Mikrofonkapsel beim Sprechen entsteht. Die Impulse unseres Computers aber sind eine Folge von Gleichströmen. Da im Telefonnetz an allen Ecken und Enden Transformatoren eingebaut sind, um bestimmte Abschnitte galvanisch voneinander zu trennen, und weil alle paar Kilometer Tonfrequenzverstärker sitzen, die Leitungsverluste ausgleichen sollen, wäre ein solcher Versuch von vornherein zum Scheitern verurteilt.

Doch nicht gleich die Flinte ins Korn werfen! Der Gedanke liegt nahe, für die beiden Zustände HIGH und LOW einfach zwei unterschiedliche Tonfrequenzen einzusetzen, die – genau wie die Sprachfrequenzen – alle Hürden unbeschadet nehmen können. Selbst die Einspeisung ins Telefonnetz stellt kein Problem mehr dar, denn man kann ja einen Lautsprecher an den Hörer halten. Am anderen Ende werden die erzeugten Frequenzen mit einer ähnlichen Anordnung wieder in ihr logisches Äquivalent umgewandelt.

Fachleute sprechen von Modulation und Demodulation der Daten. Ein Gerät, das die Umwandlung besorgt, heißt Modem (MOdulator/DEModulator) und wenn es die Tonfolgen über einen Lautsprecher zum Hörer übermittelt, ist es ein Akustikkoppler.

Im dargestellten Fall lassen sich die Informationen nur in eine Richtung übertragen. Soll gleichzeitig empfangen und

gesendet werden, dann wählt man einfach zwei weitere Frequenzen für den Rückkanal. Wegen des schon weiter oben vorgestellten Software-Protokolls zum Unterbrechen des Senders ist das wichtig, aber auch zur Eindämmung von Übertragungsfehlern. Jeder, der einmal über weite Strecken telefoniert hat, weiß, wie viel Knacken und Rauschen die Verständigung beeinträchtigen kann. Digitale Informationen sind da wesentlich empfindlicher als Sprache. Üblicherweise verfährt man deshalb folgendermaßen: Jedes übermittelte Zeichen wird vom Empfänger zurückgesendet und beim Sender mit dem Original verglichen. Nur wenn es korrekt war, wird es auf dem Bildschirm dargestellt. Andernfalls folgt ein Steuerzeichen und die Übermittlung wird wiederholt. Diese Übertragungsart nennt sich dann Vollduplex, bei einfacher Übermittlung – zwar in beiden Richtungen, aber ohne Kontrolle – handelt es sich um Halbduplex-Betrieb.

Natürlich sind die entsprechenden Frequenzen genormt, und ebenso selbstverständlich (wie sollte es anders sein?) gibt es wieder abweichende Normen in verschiedenen Ländern. Damit Sie problemlos alle Datenbanken und Mailboxen der Welt ankurbeln können, hier die Bauanleitung für einen wirklich universellen Akustikkoppler:

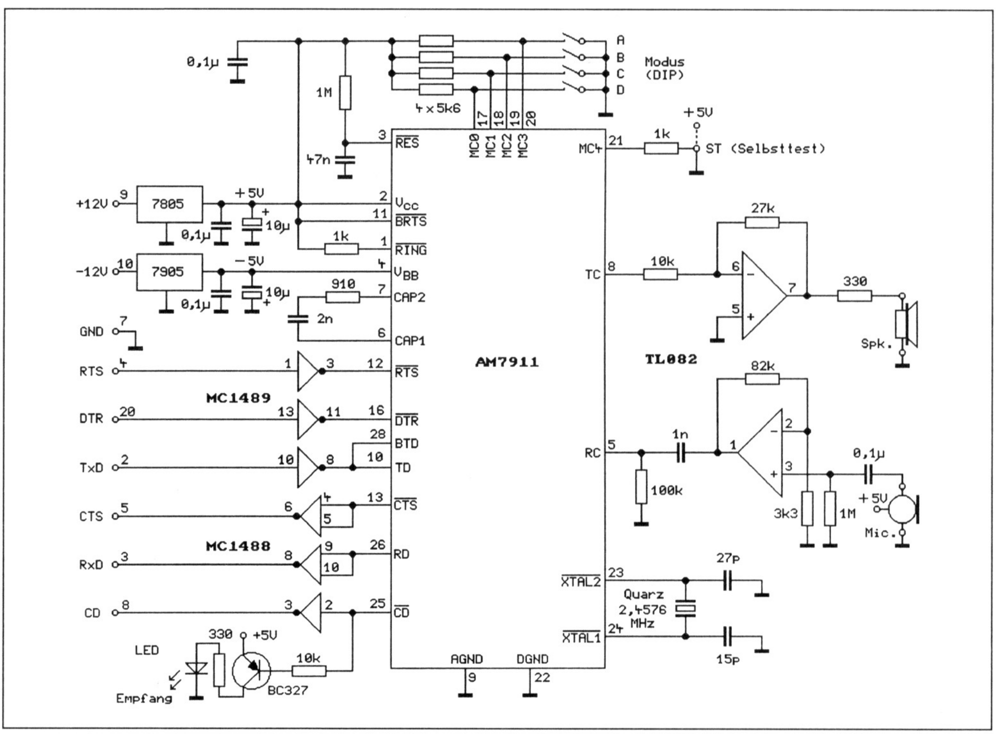


Bild 3.14: Der Modem-Schaltplan

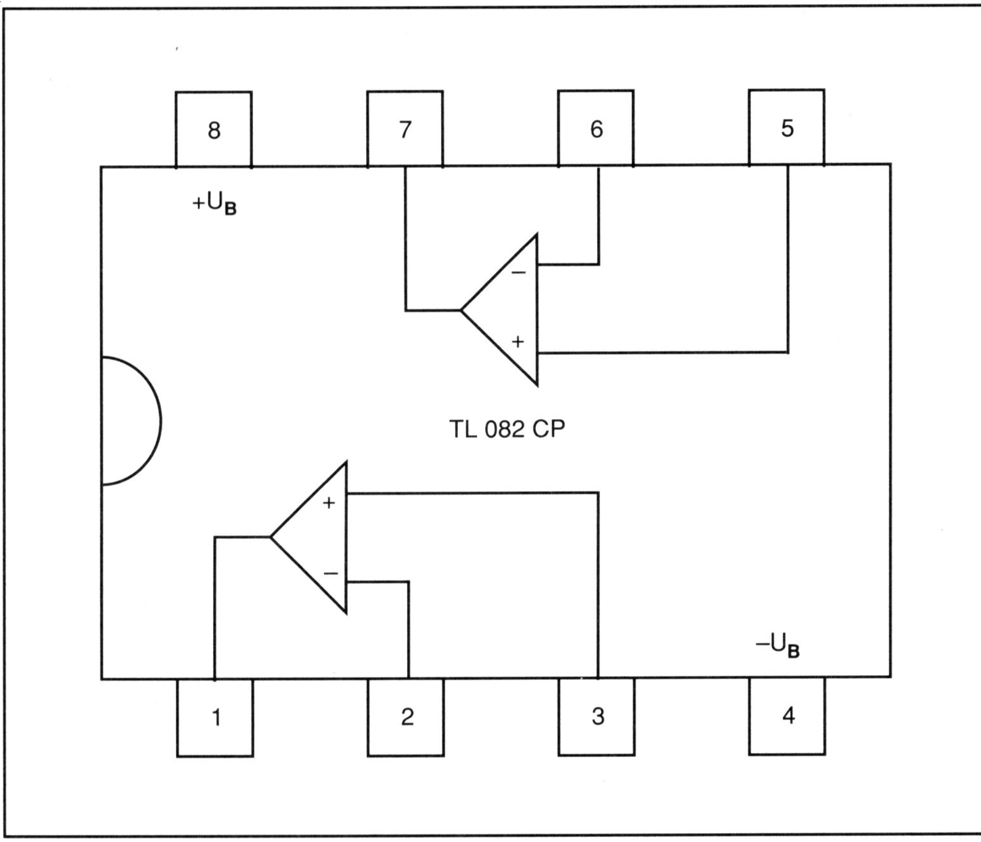


Bild 3.15: Innenschaltung des Doppel-Breitband-Operationsverstärkers mit J-FET-Eingängen TL 082 CP

3.4.4.2 Ein Weltmodem

Das Herz des Zusatzes bildet ein spezieller Modem-Chip von Advanced Micro Devices, der nur wenige externe Bauelemente benötigt. Über Dip-Schalter kann er auf die verschiedenen Normen der Welt umgestellt werden. Für die ordnungsgemäße Verbindung zum Rechner verfügt er über diverse TTL-Ein- und Ausgänge. Weil der Amiga mit RS-232-Pegeln arbeitet, wird eine Umwandlung nötig. Zu diesem Zweck dienen spezielle Bausteine vom Typ MC1488 und MC1489 von Motorola. Ebenso lassen sich auch die kompatiblen Typen 75188 und 75189 einsetzen.

Bild 3.14 zeigt den Schaltplan des Weltmodems. Zur Anpassung der analogen Signalamplituden dienen zwei Operationsverstärker, die gemeinsam in einem IC vom Typ TL082 untergebracht sind. Dabei wird die Sendefrequenz vom Pin TC auf den einen Verstärker gebracht, so daß sie an dessen Ausgang über einen kleinen Lautsprecher ausgegeben wird. Der

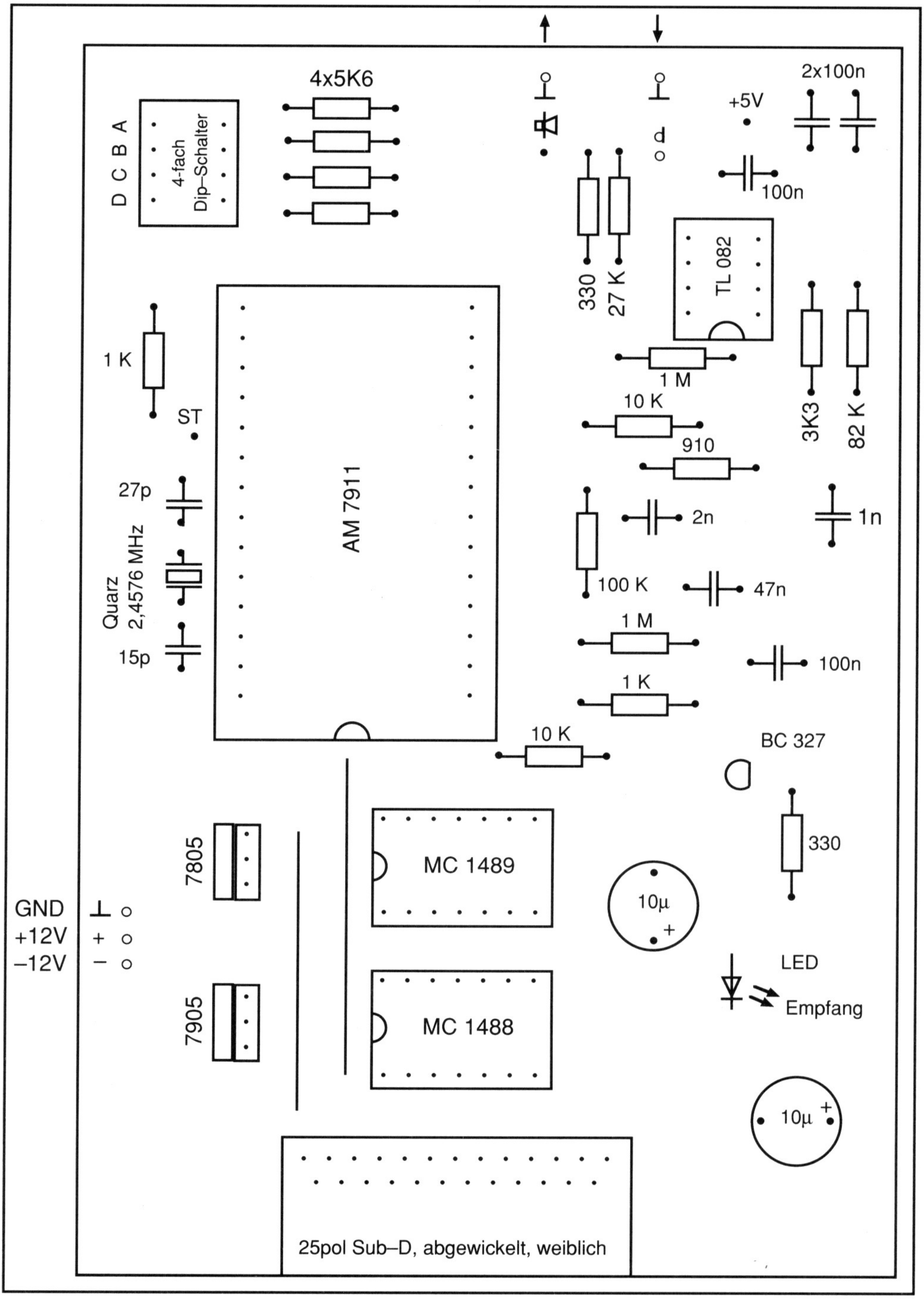


Bild 3.16: Bestückung des Modems für Amiga 500/2000

andere Weg führt von einer Kondensatormikrofonkapsel, die in jedem guten Elektronikgeschäft für wenige Mark zu haben ist, auf den zweiten Verstärker und von dessen Ausgang über einen Kondensator zum Pin RC. Für den Verstärkungsfaktor ist jeweils der Gegenkopplungswiderstand vom invertierenden Eingang zum Ausgang des Operationsverstärkers zuständig. Bei Bedarf kann er leicht geändert werden. Bild 3.15 enthält die Pinbelegung des Operationsverstärkers.

3.4.4.3 Aufbau des Modems

Bild 3.16 zeigt den Bestückungsplan für das Modem-Layout nach Bild 3.17. Für den Betrieb am Amiga 1000 gilt Bild 3.18. Richten Sie sich nach den Hinweisen im Anhang A. Achten Sie besonders auf die richtige Polung der Elkos und der Dioden. Wegen der negativen Hilfsspannung liegt der Elko hinter dem Spannungsregler 7905 mit seinem Pluspol an Masse!

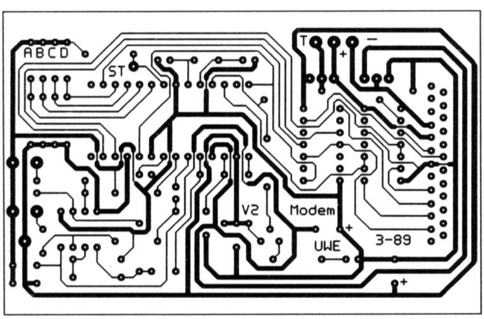


Bild 3.17: Das Modem-Layout

Am hinteren Ende der Platine befinden sich die Anschlüsse für den Lautsprecher (links) und die Kondensatormikrofonkapsel (rechts). Der Pin ganz rechts führt +5 Volt und kann zur Stromversorgung des Mikrofons dienen.

Vor allem der +5V-Spannungsregler 7805 wird im Betrieb recht warm. Soll ein Kühlblech verwendet werden, ist darauf zu achten, daß die Metallfahnen der Spannungsregler auf unterschiedlichen Potentialen liegen. Mindestens einer von beiden muß also isoliert montiert werden.

Der Amiga 1000 besitzt wie gesagt eine von den anderen Modellen abweichende Steckerbelegung. Daher muß die Verbindung zur Modemplatine über ein Kabel erfolgen. Geeignet ist der in Bild 3.19 vorgestellte Adapter. Bei den neueren Amiga-Modellen stehen am RS-232-Steckverbinder lediglich die Spannungen +12 Volt und –12 Volt zur Versorgung externer Geräte zur Verfügung. Beim Amiga 1000 sind es +5 Volt, +12 Volt und -5 Volt. Dadurch vereinfacht sich die Schaltung dort sogar ein wenig: Der Negativ-Spannungsregler 7905 kann entfallen und ist durch eine Brücke nach Bild 3.18 zu ersetzen. Das einzige Bauteil, das –12 Volt Versorgungsspannung benötigt, ist nämlich der RS-232-Pegelwandler MC1488. Er funktioniert auch, wenn ihm nur –5 Volt angeboten werden. Im Amiga 1000 selbst wird dieser Baustein ebenfalls unsymmetrisch mit +12 und –5 Volt betrieben. Die Ausgangspegel sind dann zwar auf diese Werte der Versorgungsspannung begrenzt, bei nicht allzulangen Verbindungskabeln spielt das jedoch keine Rolle, weil von der Gegenseite laut RS-232-Norm Spannungen bis –3 Volt noch sicher erkannt werden.

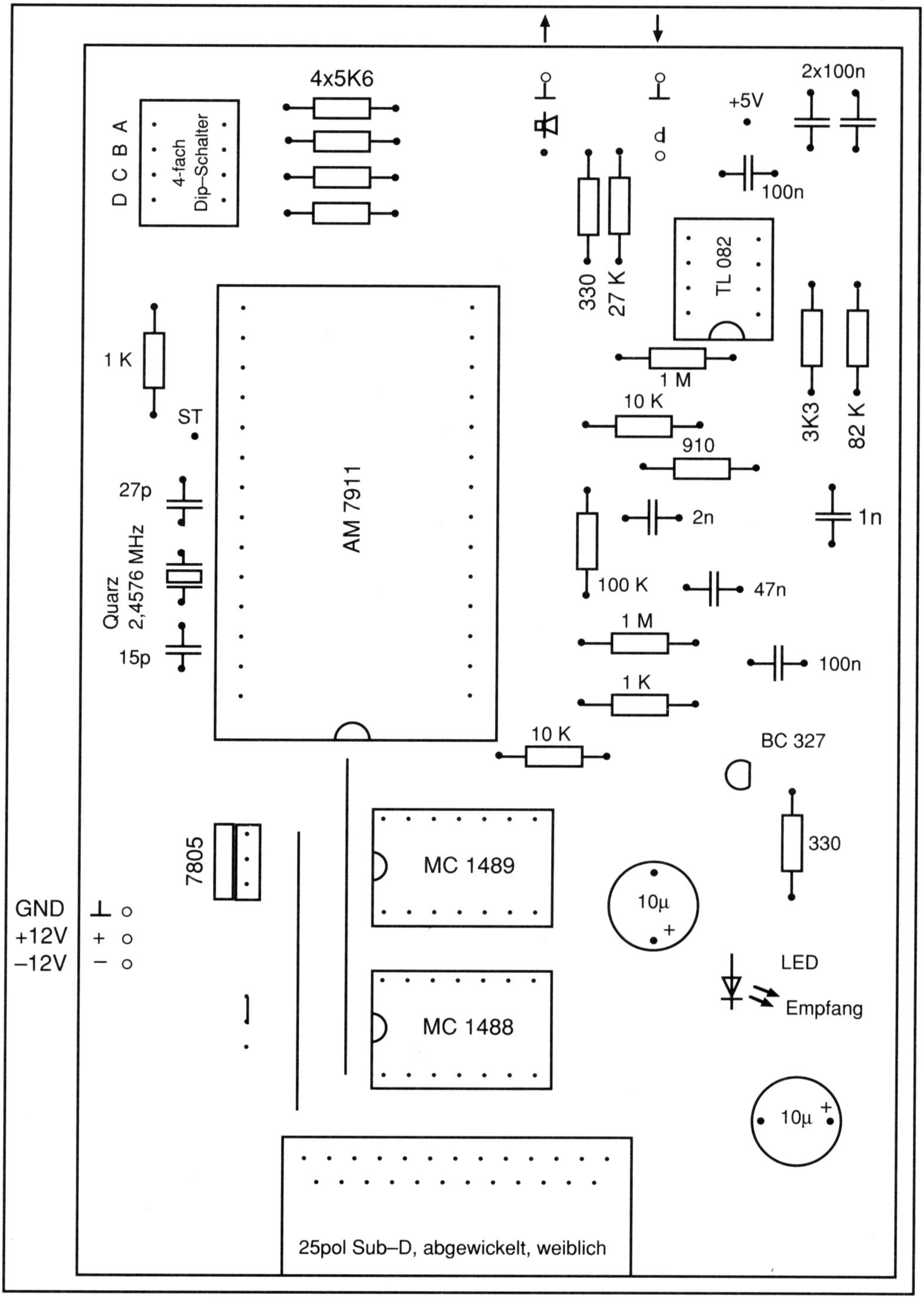


Bild 3.18: Bestückung des Modems für Amiga 1000

1	Modemchip AM7911 (bzw. AM7910)
1	IC-Sockel 28-pol
1	Zweifach-Operationsverstärker TL082
1	IC-Sockel 8-pol
1	Pegelwandler MC1488 (bzw. 75188)
1	Pegelwandler MC1489 (bzw. 75189)
2	IC-Sockel 14-pol
1	Spannungsregler 7905
1	Spannungsregler 7805 (entfällt für Amiga 1000)
1	Transistor BC327 o.ä.
1	Leuchtdiode, Farbe nach Wahl
1	Quarz 2,4576MHz (HC-18/U)
2	Widerstände 1 Megaohm
1	Widerstand 27 Kiloohm
1	Widerstand 100 Kiloohm
1	Widerstand 82 Kiloohm
2	Widerstände 10 Kiloohm
4	Widerstände 5,6 Kiloohm
1	Widerstand 3,3 Kiloohm
2	Widerstände 1 Kiloohm
1	Widerstand 910 Ohm (100 Ohm bei AM 7910)
2	Widerstände 330 Ohm
2	Elkos 10 Mikrofarad / 16 Volt, radial
4	Kondensatoren 0,1 Mikrofarad, Keramik
1	Kondensator 47 Nanofarad
1	Kondensator 2.2 Nanofarad
1	Kondensator 1 Nanofarad
1	Kondensator 27 Pikofarad
1	Kondensator 15 Pikofarad
1	Dil-Schalter, 4-fach
1	Kondensatormikrofonkapsel für 5V Betriebsspannung
1	Miniaturlautsprecher 8 Ohm (bzw. Telefonhörkapsel)
2	Installationsmuffen, NW 50/2'', Innendurchmesser etwa 60mm
1	Sub-D-Stecker, 25-pol, weiblich, abgewinkelte Kontakte
1	Platine nach Bild 3.17
9	Lötnägel

Tabelle 3.15: Die Bauteile des Weltmodems

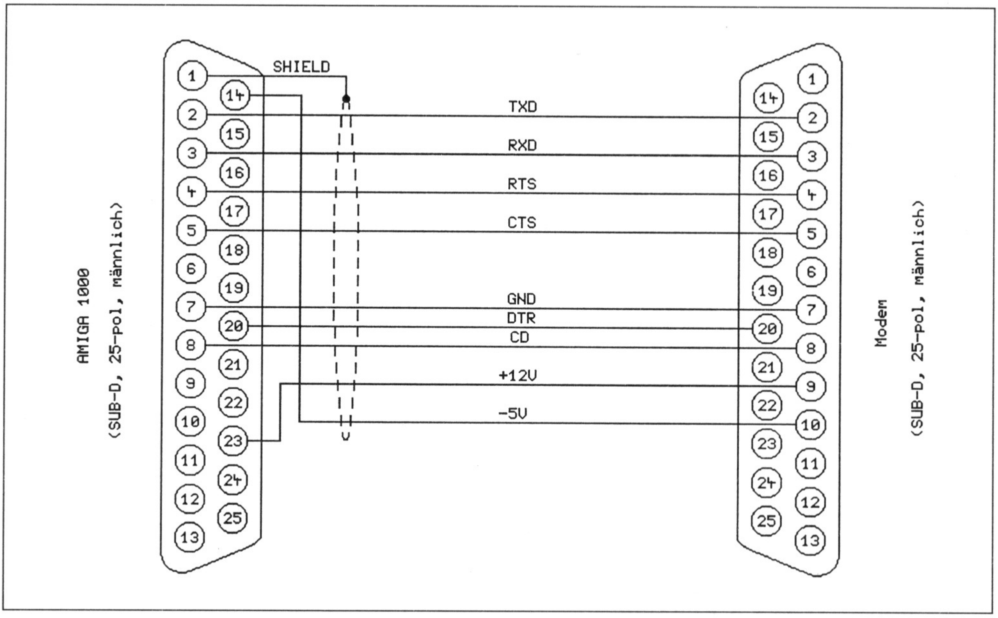


Bild 3.19: Steckerbelegung für ein Kabel zum Anschluss des Modems an einen Amiga 1000. Der Spannungsregler 7905 auf der Modemplatine muß in diesem Fall überbrückt werden.

Leider gibt es auch beim Amiga 2000 Probleme mit der Versorgungsspannung, weil im Rechner zwischen Netzteil und RS-232-Stecker mit 470 Ohm unnötig hohe Längswiderstände zur Strombegrenzung eingebaut wurden.

Am sinnvollsten ist es, die beiden Übeltäter zu überbrücken. Lesen Sie dazu Kapitel 2.1.2. Wenn Sie Ihrem Rechner nicht direkt mit dem Lötkolben zu Leibe rücken wollen, bleibt nur die Versorgung über ein externes Netzteil. Trennen Sie dazu die Verbindungen zur Spannungsversorgung auf der Modem-Platine durch (!) und schließen Sie das Netzteil (–12 Volt, +12 Volt und Masse) an die gekennzeichneten Punkte an.

3.4.4.4 Es piept!

Akustikkoppler haben bei all ihrer Genialität einen gravierenden Nachteil: Sie sind von akustischen Umwelteinflüssen abhängig. Deshalb muß man Schallquelle und -aufnehmer jeweils so dicht wie möglich zusammenbringen und auf bestmögliche akustische Abschirmung gegen Umgebungsgeräusche Wert legen. Sowohl Mikrofon als auch die Hörkapsel sollten in Gummimuffen eingebaut werden, die als Installationsbedarf für Sanitäranlagen in Baumärkten erhältlich sind. Foto 3.3 zeigt links eine geeignete Kondensatormikrofonkapsel, in der Mitte eine Telefonhörkapsel, die als Schallwandler geeignet ist, und ganz rechts eine Gummimuffe mit eingebautem Mikrofon.

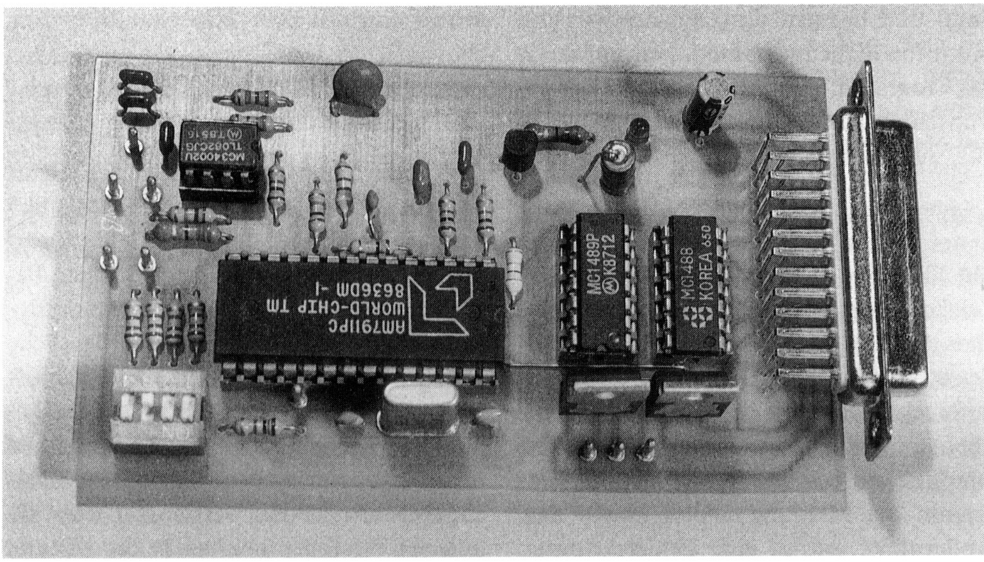


Foto 3.2

Der Einbau des Modems in ein Gehäuse lohnt sich kaum. Wegen der fehlenden Zugentlastung für die Anschlußkabel sollten diese genügend lang sein. Für das Mikrofonkabel empfiehlt sich eine abgeschirmte Ausführung, wobei das Abschirmgeflecht mit dem hinteren Lötstift (Masse) verbunden werden sollte.

Falls Sie den AM7911 nicht bekommen, können Sie auch den AM7910 verwenden, der lediglich einige Übertragungsnormen weniger beherrscht. In diesem Fall sollte aber der Widerstand an Pin 7 (910 Ohm) gegen einen mit 100 Ohm ausgetauscht werden.

Foto 3.2 zeigt das fertig aufgebaute Modem in der Version für den Amiga 500/2000.

Die Platine enthält zwischen Quarz und Dip-Schaltern einen mit ST bezeichneten Testpunkt. Schließt man ihn mit +5 Volt

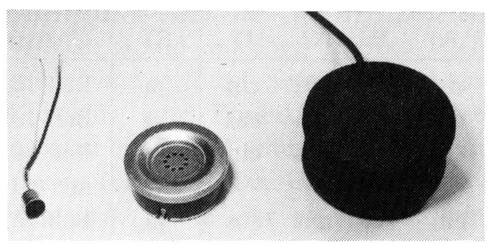


Foto 3.3: Elektroakustische Wandler

(zum Beispiel von der Mikrofon-Stromversorgung) kurz, gelangt der Modemchip in einen Selbsttest-Modus, bei dem der Eingangszweig auf die Sendefrequenz reagiert. Schließen Sie die Modem-Schaltung an den RS-232-Port Ihres Amiga an und schalten Sie den Rechner ein. Der Lautsprecher bleibt zunächst stumm. Ein Selbsttest ist nur bei den in der Spalte ST von Tabelle 3.16 mit »ja« gekennzeichneten Dip-Schalter-Kombinationen möglich. Wählen Sie also ein geeignetes Übertragungsformat mit 300 oder 600

Baud und bringen Sie Lautsprecher und Mikrofon dicht zusammen. Nun rufen Sie das Programm MODEMTEST mit der gewünschten Übertragungsgeschwindigkeit auf, zum Beispiel

```
MODEMTEST 300
```

für 300 Baud. Daraufhin wird eine Test-Zeichenfolge gesendet, was im Lautsprecher deutlich hörbar sein muß. Währenddessen sollte die LED am Modem aufleuchten. Konnte die Test-Zeichenfolge über das Mikrofon wieder einwandfrei empfangen werden, gibt das Testprogramm die Meldung »Alles klar!« aus. Andernfalls erfolgt eine Fehlermeldung. Für den normalen Betrieb bleibt der Testpin ST nur mit dem Widerstand beschaltet.

3.4.4.5 Das Modem im Einsatz

Als Selbstbau-Modem hat unser Gerät natürlich keine FTZ-Nummer und darf daher in der Bundesrepublik Deutschland nicht am öffentlichen Fernmeldenetz betrieben werden. Im Bundespostministerium wurde jedoch bereits eine deutliche Lockerung der Bestimmungen beschlossen, so daß in nächster Zeit mit der allgemeinen Betriebsgenehmigung für alle Akustikkoppler zu rechnen ist. Eine solche Regelung scheint vernünftig, denn ob nun ein Telefonteilnehmer in die Leitung zirpt, oder ein Gerät, bleibt sich doch

A	B	C	D	ST	Übertragungsart
ein	ein	ein	ein	ja	Bell 103 Originate 300 Baud Vollduplex
ein	ein	ein	aus	ja	Bell 103 Answer 300 Baud Vollduplex
ein	ein	aus	ein	ja	Bell 202 1200 Baud Halbduplex mit 5 Baud Rückkanal
ein	ein	aus	aus	ja	Bell 202 1200 Baud Halbduplex mit Amplituden-Equalizer und 5 Baud Rückkanal
ein	aus	ein	ein	ja	CCITT V.21 Originate 300 Baud Vollduplex
ein	aus	ein	aus	ja	CCITT V.21 Answer 300 Baud Vollduplex
ein	aus	aus	ein	ja	CCITT V.23 Modus 2 1200 Baud Halbduplex
ein	aus	aus	aus	ja	CCITT V.23 Modus 2 mit Amplituden-Equalizer 1200 Baud Halbduplex
aus	ein	ein	ein	ja	CCITT V.23 Modus 1 600 Baud Halbduplex
aus	ein	aus	ein	ja	Bell 202 1200 Baud mit 150 Baud Rückkanal
aus	ein	aus	aus	--	Bell 202 1200 Baud mit Amplituden-Equalizer und 150 Baud Rückkanal
aus	aus	ein	ein	--	CCITT V.23 Mode 1 600 Baud mit Soft Turn-Off
aus	aus	aus	ein	--	CCITT V.23 Mode 2 1200 Baud mit Soft Turn-Off
aus	aus	aus	aus	--	CCITT V.23 Mode 2 1200 Baud mit Amplituden-Equalizer und Soft Turn-Off

Tabelle 3.16: Übertragungsnormen des AM7911

gleich, solange nicht direkt in die Fernmeldeanlagen eingegriffen wird. Das hier vorgestellte Modem wurde als Akkustikkoppler konzipiert und ist daher auch nicht direkt an das Netz anschließbar.

Es sollte selbstverständlich sein, daß unser Modem nur bei ausgeschaltetem Computer aufgesteckt werden darf. Danach ist es bereits betriebsfertig.

Nun laden Sie ein Terminalprogramm und stellen die Parameter der Gegenseite ein. In den Rufnummernverzeichnissen der Mailboxen oder Datenbanken sind meist auch alle Übertragungsparameter angegeben. Normalerweise werden das 8 Datenbits, keine Parität, 1 Stopbit sein. Als Übertragungsgeschwindigkeit ist hierzulande meist CCITT V.21, Originate 300 Baud und Vollduplex üblich. Dabei sind auf der Modem-Platine alle Dip-Schalter außer B geschlossen. Tabelle 3.16 zeigt die verschiedenen einstellbaren Möglichkeiten.

Wird das Terminalprogramm aktiviert, ertönt ein Ton aus dem Lautsprecher. Bei leichtem Klopfen auf das Mikrofon muß die Signal-LED kurz aufblitzen. Wählen Sie nun die Nummer einer Mailbox oder Datenbank und warten auf das Herstellen der Verbindung. Sobald Sie den Sendeton der Gegenstelle hören, stülpen Sie die Gummimuffe mit dem eingebauten Mikrofon auf die Hörseite und die mit dem Lautsprecher auf die Sprechseite des Handapparates.

Die LED sollte aufleuchten, zum Zeichen, daß eine Gegenstation empfangen wird. Jetzt kann der Datenaustausch beginnen.

3.4.5 Veni Vidi MIDI

Elektronische Musikinstrumente bieten eine geradezu unerschöpfliche Klangvielfalt. Allerdings erfordert es vor allem bei Synthesizern ein gutes Stück feinfühliger Arbeit, bis ein neuer Sound von Hand eingestellt ist. So waren Live-Konzerte lange Zeit unglaublich aufwendig, denn für jeden Klang stand ein komplettes Tasteninstrument auf der Bühne, das vorher in langwieriger Kleinarbeit exakt eingestellt worden war und sich nur eingeschränkt umprogrammieren ließ. Im Zeitalter der Computer sieht die Sache anders aus.

3.4.5.1 Computer im Musikgeschäft

1982 haben sich Vertreter der führenden Instrumente-Hersteller (SCI, Roland, Yamaha, Kawai und Korg) an einen Tisch gesetzt und sich auf einen Standard zur Kopplung ihrer Produkte geeinigt. Sie nannten ihn MIDI (Musical Instrument Digital Interface). Der Erfolg war überwältigend. Heute gibt es beinahe kein elektronisches Instrument mehr, das nicht die drei markanten runden Buchsen aufweist.

Aber MIDI ist nicht nur eine genormte Anschlußbuchse, sondern ein komplettes digitales Kommunikationssystem. Theoretisch benötigt man nur noch eine einzige Klaviatur, ein sogenanntes Masterkeyboard, das selbst keine Töne erzeugen kann, jedoch über MIDI verschiedene Expander steuert, zum Beispiel Synthis ohne Tastatur, die nur über MIDI spielbar sind. In der Praxis ist MIDI schon längst zum Zauberwort der elektronischen Musik geworden. Immer mehr Instrumente werden

mit den notwendigen technischen Voraussetzungen versehen. Neben Klaviaturen können inzwischen auch elektrische Gitarren, Baßgitarren, Orgeln und Akkordeons als Master-Instrumente eingesetzt werden. Die Pitchrider-Schnittstelle der Firma IVL verwandelt sogar per Mikrofon aufgenommene Töne in MIDI-Signale.

Durch einfachen Knopfdruck können die Expander über einen ebenfalls per MIDI angeschlossenen Computer sehr schnell mit neuen Klängen geladen werden.

3.4.5.2 Das MIDI-Datenformat

Als ein solcher Steuercomputer mit Massenspeicher bietet sich der Amiga geradezu an. Es existieren bereits einige MIDI-fähige Programme, wie etwa Aegis Sonix, Deluxe Music Construction Set und Soundscape Pro MIDI Studio. Allerdings muß noch ein kleiner Adapter angefertigt werden, um den Amiga MIDI-fähig zu machen. Grundsätzlich folgt die MIDI-Übertragung der RS-232-Norm. Die Daten werden seriell übertragen. Jede Informationseinheit besteht aus einem Startbit, acht Datenbits und einem Stopbit. Logisch 0 bedeutet dabei Strom ein. Die Übertragungsrate wurde auf 31,25 Kilobaud festgelegt. Dieser auf den ersten Blick etwas ungewöhnliche Wert ergibt sich aus der Teilung eines 1-MHz-Taktes durch 32. Das ist sinnvoll, denn die meisten MIDI-Geräte laufen nun einmal mit ganzen Vielfachen dieser Frequenz. Computer mit Bildschirm werden oft – wie auch der Amiga – mit Vielfachen der Video-Frequenz betrieben und haben bei der Erzeugung der MIDI-Datenrate so ihre Schwierigkeiten. Durch die flexibel programmierbare Hardware läßt den Amiga diese Klippe aber kalt.

Trotz der digitalen Ansteuerung bleibt die vom Synthesizer generierte Tonfrequenz in den allermeisten Fällen heute noch analog und wird auch analog verstärkt. Dabei machen sich bereits geringe Störspannungen als unangenehme Nebengeräusche bemerkbar. Gerade digitale Geräte sind besonders starke Störquellen. Durch schnell ansteigende Spannungen produzieren sie unkontrollierbare Oberwellen. Daher muß auf eine gute Abschirmung sowohl der Geräte selbst wie auch ihrer Verbindungskabel geachtet werden. Bei jedem Schaltvorgang wird aber auch die Stromversorgung des digitalen Gerätes kurzzeitig stärker belastet. Aufgrund der Zuleitungswiderstände entsteht dabei ein leicht schwankendes Massepotential auf der Platine und natürlich auch an den Steckverbindungen. Bei einer direkten Verbindung mit einem analogen Verstärker würde sich das als starkes Sirren unangenehm bemerkbar machen. Daher werden nach der MIDI-Norm alle Geräte streng galvanisch voneinander getrennt. Die Datenübertragung geschieht über Optokoppler im Eingang jedes Gerätes. Natürlich darf hier auch die Abschirmung des Verbindungskabels nicht mit der Empfängermasse verbunden werden, da sie ja bereits im Sender angeschlossen ist. Andernfalls ergäbe sich neben der bereits beschriebenen Effekte auch noch die Gefahr einer Brummschleife durch mehrfache Erdung. Der Strom zum Betrieb der Leuchtdiode im Optokoppler kommt vollkommen aus dem Sender. Man spricht auch von einer 5mA-Stromschleife.

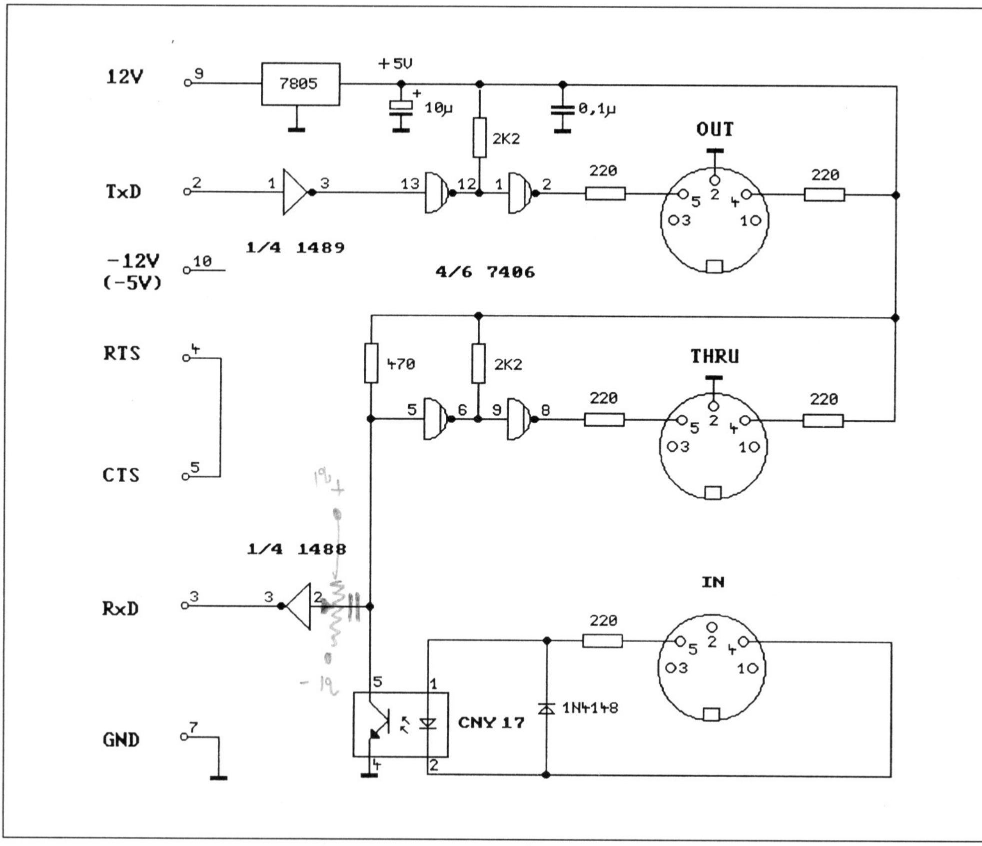


Bild 3.20: Schaltplan des Amiga-MIDI-Interfaces

Bild 3.20 zeigt den Schaltplan für ein MIDI-Interface am Amiga, das den Standard voll erfüllt. Über die Buchse OUT können Daten an einen MIDI-Empfänger übermittelt werden. Die seriellen Spannungsimpulse kommen über einen Pegelwandler MC1489 direkt von der RS-232-Schnittstelle des Amiga. Der Open-Collector-Treiber 7406 sorgt dabei zusammen mit den Widerständen für einen optimalen Betrieb des dort befindlichen Optokopplers.

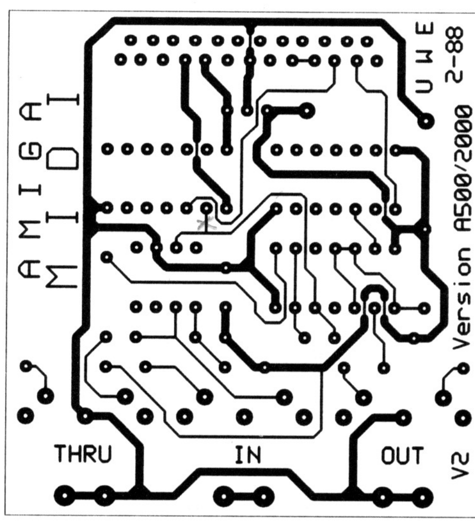


Bild 3.21: Layout des MIDI-Interfaces

Über IN werden Daten von einem externen MIDI-Sender mit Hilfe des erwähnten Optokopplers zum Eingang der Amiga-RS-232-Schnittstelle übertragen. Der Schnittstellentreiber MC1488 dient dabei zur Umwandlung der TTL-Signale in RS-232-Pegel.

Gleichzeitig werden die Daten an der Buchse THRU ähnlich wie beim Sendekanal wieder unverändert für ein nachgeschaltetes Gerät zur Verfügung gestellt. Weil so mehrere Geräte einfach hintereinandergeschaltet werden können, spricht man auch vom MIDI-Bus.

Die Amiga-MIDI-Schnittstelle ist mit dem Layout aus Bild 3.21 schnell aufgebaut. Tabelle 3.17 führt alle benötigten Bateile auf. Sie werden nach dem Bestückungs-

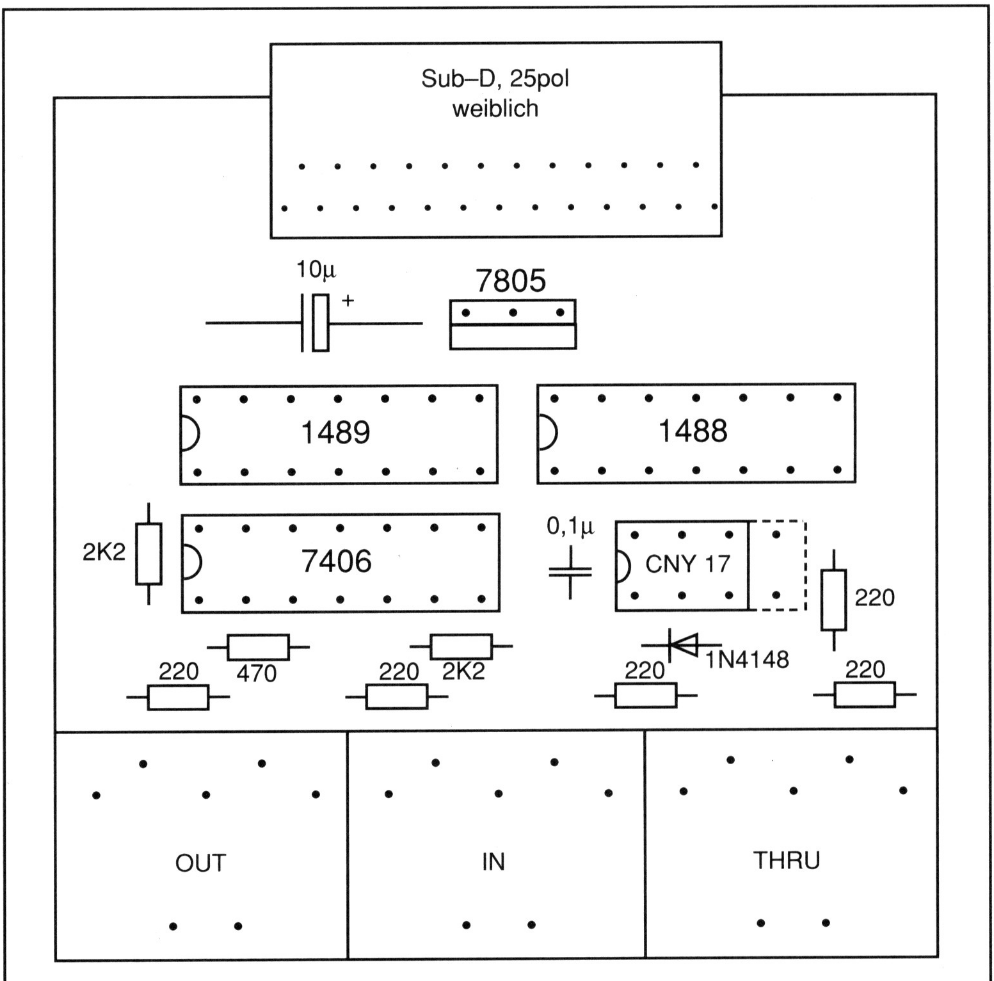


Bild 3.22: Bestückung der MIDI-Platine

1	Pegelwandler MC1488
1	Pegelwandler MC1489
1	Open-Collector-Treiber 7406
1	Optokoppler CNY17
1	Spannungsregler 7805
3	IC-Sockel, 14-pol
1	IC-Sockel, 6-pol bzw. 8-pol
1	Diode 1N4148
5	Widerstände 220 Ohm
1	Widerstand 470 Ohm
2	Widerstände 2,2 Kiloohm
1	Kondensator 10 Mikrofarad
1	Kondensator 0,1 Mikrofarad
1	Sub-D-Stecker, 25-pol, weiblich
3	DIN-Buchsen, 5-pol für Printmontage
1	Platine nach Bild 3.21

Tabelle 3.17: Bauteile für die MIDI-Schnittstelle

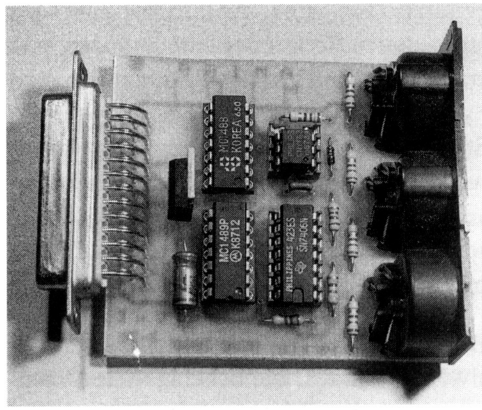


Foto 3.4: MIDI-Interface

plan in Bild 3.22 verlötet. Richten Sie sich nach den Hinweisen in Anhang A. Der Spannungsregler wird mit seiner Kühlfahne zu den MIDI-Buchsen hin eingesetzt.

An den Amiga 1000 muß die Platine wegen der anderen Steckerbelegung über das in Kapitel 3.4.2.2 vorgestellte Adapterkabel angeschlossen werden. Wie schon beim Modem in Kapitel 3.4.4 beschrieben, versieht der Pegelwandler MC1488 seinen Dienst auch mit einer negativen Versorgungsspannung von nur –5 statt –12 Volt. Foto 3.4 zeigt das fertige MIDI-Interface.

3.4.5.3 Musik-Pipeline

Wichtiger Bestandteil des MIDI-Standards ist natürlich auch die Bedeutung der übermittelten Informationen. Bei gesetztem achten Datenbit handelt es sich um ein sogenanntes Statusbyte, das eins von 128 möglichen MIDI-Kommandos einleitet. Ihm können grundsätzlich beliebig viele ergänzende Datenbytes folgen, bei denen das achte Bit immer 0 ist. Bei Statusbytes unterscheidet man noch einmal zwischen Kanalinformationen und Systeminformationen.

Kanalinformationen bilden Kommandos für ein einziges Gerät, wobei die niederwertigen vier Bit jeweils eins von 16 am Bus anschließbaren Geräten adressieren. Die restlichen Bit des Statusbyte kennzeichnen Befehle, wie etwa »Note an« oder »Note aus«, »Tonhöhe«, »Anschlagsdynamik« oder »Klangart«, deren genaue Parameter in den folgenden ein oder zwei Datenbyte übertragen werden. Die Zuordnung dieser Befehle zu bestimmten Schaltern und Reglern ist herstellerspezifisch, also von Gerät zu Gerät verschieden.

Systeminformationen beziehen sich nicht auf einen bestimmten Kanal und lassen sich in drei Untergruppen aufteilen. Die Common-Kommandos haben Einfluß auf alle angeschlossenen Geräte. Sie werden

Über IN werden Daten von einem externen MIDI-Sender mit Hilfe des erwähnten Optokopplers zum Eingang der Amiga-RS-232-Schnittstelle übertragen. Der Schnittstellentreiber MC1488 dient dabei zur Umwandlung der TTL-Signale in RS-232-Pegel.

Gleichzeitig werden die Daten an der Buchse THRU ähnlich wie beim Sendekanal wieder unverändert für ein nachgeschaltetes Gerät zur Verfügung gestellt. Weil so mehrere Geräte einfach hintereinandergeschaltet werden können, spricht man auch vom MIDI-Bus.

Die Amiga-MIDI-Schnittstelle ist mit dem Layout aus Bild 3.21 schnell aufgebaut. Tabelle 3.17 führt alle benötigten Bateile auf. Sie werden nach dem Bestückungs-

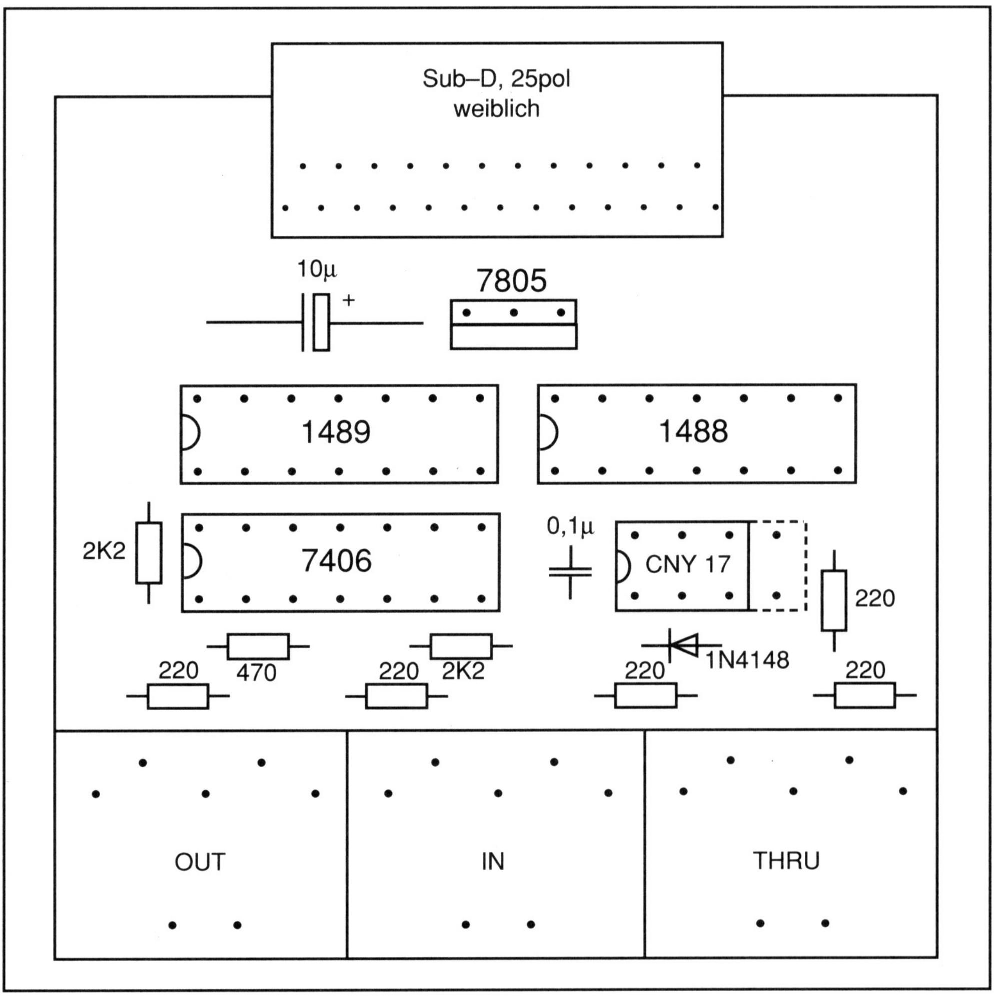


Bild 3.22: Bestückung der MIDI-Platine

1	Pegelwandler MC1488
1	Pegelwandler MC1489
1	Open-Collector-Treiber 7406
1	Optokoppler CNY17
1	Spannungsregler 7805
3	IC-Sockel, 14-pol
1	IC-Sockel, 6-pol bzw. 8-pol
1	Diode 1N4148
5	Widerstände 220 Ohm
1	Widerstand 470 Ohm
2	Widerstände 2,2 Kiloohm
1	Kondensator 10 Mikrofarad
1	Kondensator 0,1 Mikrofarad
1	Sub-D-Stecker, 25-pol, weiblich
3	DIN-Buchsen, 5-pol für Printmontage
1	Platine nach Bild 3.21

Tabelle 3.17: Bauteile für die MIDI-Schnittstelle

plan in Bild 3.22 verlötet. Richten Sie sich nach den Hinweisen in Anhang A. Der Spannungsregler wird mit seiner Kühlfahne zu den MIDI-Buchsen hin eingesetzt.

An den Amiga 1000 muß die Platine wegen der anderen Steckerbelegung über das in Kapitel 3.4.2.2 vorgestellte Adapterkabel angeschlossen werden. Wie schon beim Modem in Kapitel 3.4.4 beschrieben, versieht der Pegelwandler MC1488 seinen Dienst auch mit einer negativen Versorgungsspannung von nur –5 statt –12 Volt. Foto 3.4 zeigt das fertige MIDI-Interface.

3.4.5.3 Musik-Pipeline

Wichtiger Bestandteil des MIDI-Standards ist natürlich auch die Bedeutung der übermittelten Informationen. Bei gesetztem achten Datenbit handelt es sich um ein sogenanntes Statusbyte, das eins

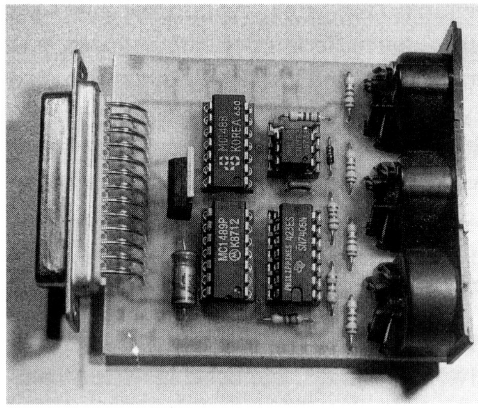


Foto 3.4: MIDI-Interface

von 128 möglichen MIDI-Kommandos einleitet. Ihm können grundsätzlich beliebig viele ergänzende Datenbytes folgen, bei denen das achte Bit immer 0 ist. Bei Statusbytes unterscheidet man noch einmal zwischen Kanalinformationen und Systeminformationen.

Kanalinformationen bilden Kommandos für ein einziges Gerät, wobei die niederwertigen vier Bit jeweils eins von 16 am Bus anschließbaren Geräten adressieren. Die restlichen Bit des Statusbyte kennzeichnen Befehle, wie etwa »Note an« oder »Note aus«, »Tonhöhe«, »Anschlagsdynamik« oder »Klangart«, deren genaue Parameter in den folgenden ein oder zwei Datenbyte übertragen werden. Die Zuordnung dieser Befehle zu bestimmten Schaltern und Reglern ist herstellerspezifisch, also von Gerät zu Gerät verschieden.

Systeminformationen beziehen sich nicht auf einen bestimmten Kanal und lassen sich in drei Untergruppen aufteilen. Die Common-Kommandos haben Einfluß auf alle angeschlossenen Geräte. Sie werden

genutzt, um eine Anfangsstellung für einen neuen Song oder eine Sequenz auszuwählen. Die Echtzeit-Kommandos sind Synchronisationssignale zum Starten, Stoppen und Takten von Rhythmusmaschinen und Sequencern. Sie enthalten keine Datenbytes und können jederzeit gesendet werden, auch zwischen den Datenbytes eines anderen Kommandos.

Die dritte Untergruppe bilden die Exklusiv-Kommandos, die beliebig viele Datenbytes enthalten können. Dies können zum Beispiel komplexe Sound-Parameter oder ganze Sound-Bänke sein, die über MIDI zu oder von einem Computer gesendet werden, der sie speichert oder verändert. Natürlich sind auch diese Daten von Gerät zu Gerät unterschiedlich. Generell ignoriert aber ein MIDI-Gerät einfach alle Daten, die es nicht interpretieren kann, so daß es nicht zu größeren Störungen beim Auswechseln von Geräten kommt und die Universalität gewahrt bleibt.

3.5 Augenblick mal

Gerade bei der seriellen Schnittstelle war sehr oft die Rede von Interrupt-Bits. Daher soll hier der Interrupt-Controller in PAULA genauer beschrieben werden.

3.5.1 Interrupt Controller

Der Prozessor 68000 bietet insgesamt sieben verschiedene Interruptebenen an. Dazu verfügt er über drei Interrupt-Eingangsleitungen, bezeichnet mit $\overline{IPL0}$, $\overline{IPL1}$ und $\overline{IPL2}$. Die Balken über den Signalnamen deuten bereits negative Logik an. Falls alle drei Leitungen gleichzeitig HIGH führen, ist keine Interruptanforderung aktiv, und der Prozessor geht ungestört seiner Arbeit nach. Jeder andere Zustand signalisiert aber einen Unterbrechungswunsch der prozessorexternen Hardware, wobei jede einzelne Kombination eine der sieben Interruptebenen ausgewählt. Der Prozessor arbeitet beim Eintreffen einer solchen Bedingung den gerade begonnenen Maschinenbefehl noch fertig ab und verzweigt dann zu einer für jede Ebene spezifischen Adresse, bei der die Abarbeitung des jeweiligen Interruptprogramms beginnt.

Der »Peripheriebaustein« PAULA kontrolliert die drei genannten Leitungen, indem er die Signale von 14 unterschiedlichen Interruptquellen verwaltet, jedem eine feste Ebene zuordnet und die Unterdrückung (Maskierung) einzelner Unterbrechungsanforderungen ermöglicht. Bei den Interruptquellen handelt es sich vor

Register	Adresse	R/W		
INTREQ	$DFF09C	W	Interrupt Request Bits	schreiben
INTREQR	$DFF01E	R	Interrupt Request Bits	lesen
INTENA	$DFF09A	W	Interrupt Enable Bits	schreiben
INTENAR	$DFF01C	R	Interrupt Enable Bits	lesen

Tabelle 3.18: Die Register zur Steuerung der Interrupts

allem um interne Register. Lediglich die drei Signale $\overline{\text{INT2}}$, $\overline{\text{INT3}}$ und $\overline{\text{INT6}}$ werden über Pins von anderen Schaltungsteilen zugeführt. Die Namen sind nach der Ebene des auszulösenden Interrupts gewählt. $\overline{\text{INT3}}$ kommt von AGNUS, $\overline{\text{INT2}}$ vom I/O-Baustein 8520-B und $\overline{\text{INT6}}$ vom 8520-A.

Zur Programmierung des Interrupt-Controllers existieren zwei Register, die unter einer Adresse gelesen, jedoch unter einer anderen beschrieben werden können. Tabelle 3.18 gibt Auskunft.

Die INTREQ (Interrupt Request)-Register enthalten die Zustandsbits der einzelnen Interruptquellen. Normalerweise sind alle Bits gelöscht. Tritt in der Hardware eine Interruptbedingung auf, wird das zugehörige Bit in diesem Register gesetzt. Ob gleichzeitig auch ein Interrupt ausgelöst wird, entscheidet das entsprechende Bit im Interrupt Enable-Register (INTENA).

Ist es gelöscht, wird kein Interrupt ausgelöst. Die Bits im Interrupt Enable-Register bilden also eine Maske, die nur die gewünschten Quellen zum Prozessor wei-

Bit	Funkt.	Pr.	Bedeutung
15	SET/ CLEAR		Legt fest, ob gesetzte Bits die entsprechenden Stellen löschen oder setzen sollen. (Gelöschte Bits ändern nichts)
14	INTEN		Master Interrupt (nur Enable)
13	EXTER	6	Externer Interrupt
12	DSKSYN	5	Disk-Synchronisationsregister DSKSYNC hat Daten
11	RBF	5	Empfangspuffer des seriellen Ports voll
10	AUD3	4	Audio Kanal 3 hat Block beendet
9	AUD2	4	Audio Kanal 2 hat Block beendet
8	AUD1	4	Audio Kanal 1 hat Block beendet
7	AUD0	4	Audio Kanal 0 hat Block beendet
6	BLIT	3	Blitter fertig
5	VERTB	3	Start der vertikalen Bildaustastlücke
4	COPER	3	Copper
3	PORTS	2	I/O-Ports und Timer
2	SOFT	1	Reserviert für Software-Interrupt
1	DSKBLK	1	Disk-Block fertig
0	TBE	1	Serieller Port Sendepuffer leer

Tabelle 3.19: Die Bits in den Interruptregistern

termeldet. Die Interruptanforderungen werden dadurch »maskiert«.

Tabelle 3.19 zeigt die für Lesen und Schreiben identische Bedeutung der Registerbits.

Die Enable- bzw. Interrupt-Bits können vom Programm gesetzt oder gelöscht werden. Allerdings ist während eines Zugriffs nur entweder Setzen oder Löschen möglich. Bit 15 entscheidet, ob gesetzt (1) oder gelöscht (0) werden soll. Die übrigen Stellen legen durch 1-Bits die zu beeinflussenden Stellen fest. 0-Bits lassen die entsprechenden Stellen unbeeinflußt. Zum Setzen von Bit 2 muß beispielsweise der Dualwert

```
15             8  7              0

1 0 0 0 0 0 0 0   0 0 0 0 0 1 0 0
```

geschrieben werden, zum Löschen dieses Bits entsprechend der Wert

```
15             8  7              0

0 0 0 0 0 0 0 0   0 0 0 0 0 1 0 0
```

Ist irgendwann ein Bit sowohl in INTREQ wie auch in INTENA gesetzt, so wird automatisch auch Bit 14 Interrupt-Request-Register gesetzt und gleichzeitig über die drei IPL-Leitungen eine Interruptanforderung für die zugehörige Ebene zum Prozessor signalisiert.

Im Interrupt-Enable-Register dient Bit 14 als Master Interrupt Enable-Bit. Ist es gelöscht, werden alle Interruptanforderungen unterdrückt. Diese Einrichtung läßt sich sinnvoll nutzen, um für zeitkritische Programmteile alle Unterbrechungsmöglichkeiten kurzzeitig abzuschalten.

Der Prozessor kann auch durch unmittelbares Setzen eines Bits im Interrupt-Request-Register einen Interrupt softwaremäßig hervorrufen. Der Zugriff dort geschieht auf die gleiche Weise wie eben beschrieben.

Achtung: Nach der Abarbeitung der entsprechenden Interruptroutine muß im Gegensatz zur Handhabung bei den I/O-Bausteinen 8520 immer das auslösende Bit im Interrupt-Request-Register wieder gelöscht werden!

3.5.2 Interruptquellen

Die einzelnen Bedingungen, unter denen das jeweilige Bit im Interrupt-Request-Register gesetzt wird, sollen hier kurz zusammengefaßt werden:

Bit 13: Externer Interrupt Ebene 6

Dieses Signal kommt vom I/O-Baustein 8520-A. Dort gibt es wiederum mehrere Interruptquellen, die sich vormaskieren lassen. Im einzelnen:

- Unterlauf von Timer A
  z.B. bei Betrieb des seriellen Ports über die Pins POUT und BUSY der Parallelschnittstelle
- Unterlauf von Timer B
  z.B. beim Einsatz als Beam Follower zur Synchronisation des Blitters auf den Video-Teil
- TOD-Alarm
  Horizontalimpulse haben festgelegte Anzahl erreicht
- Serieller Port voll/leer
  nur bei Betrieb des seriellen Ports über

die Pins POUT und BUSY der Parallelschnittstelle

- negative Flanke an $\overline{\text{FLAG}}$ aufgetreten
  Disk Index-Signal aufgetreten

Bit 12: Disksynchronisation Ebene 5

Bit 12 von DSKSYN: Zeigt an, daß der Disk-Controler eine Sync-Marke erkannt hat.

Bit 11: Serieller Port Ebene 5

Bit 14 von SERDATR: Der Empfangspuffer des UART-Controlers enthält ein »fertig« empfangenes Byte.

Bit 10: Audio Kanal 3 Ebene 4

Signalisiert Datenblock über Kanal 3 ausgegeben, wenn der Audio-DMA im Automatikbetrieb arbeitet. Das Interrupt-Bit wird gesetzt, sobald auf das letzte Wort eines Datenblocks zugegriffen wurde.

Im Manual-Mode tritt der Interrupt auf, wenn das Audio-Datenregister für Kanal 3 zur Aufnahme eines neuen Datenwortes bereit ist.

Bit 9: Audio Kanal 2 Ebene 4

Entspricht Bit 10, jedoch für Audio-Kanal 2

Bit 8: Audio Kanal 1 Ebene 4

Entspricht Bit 10, jedoch für Audio-Kanal 1

Bit 7: Audio Kanal 0 Ebene 4

Entspricht Bit 10, jedoch für Audio-Kanal 0

Bit 6: Blitter-Interrupt Ebene 3

Zeigt an, daß der Blitter die ihm aufgetragene Arbeit erledigt hat und für eine neue Programmierung zur Verfügung steht.

Bit 5: Bildaustastlücke Ebene 3

Bit 5 von Register VERTB. Zeile 0 eines neuen Bildes wurde geschrieben. Dieser Interrupt erscheint 50mal in der Sekunde. Während der Austastlücke, also von Zeile 0 bis etwa Zeile 20, ist der Videostrahl noch nicht auf dem Bildschirm zu sehen. Daher läßt sich der Bildinhalt während dieser Zeit ändern, ohne daß im Bild Störungen zu sehen sind. Ebenso lassen sich viele andere systemspezifische Aufgaben während dieser Zeit ausführen, zum Beispiel die Umprogrammierung des Coppers und anderer Kontrollparameter.

Bit 4: Copper-Interrupt Ebene 3

Bit 4 vom Register COPPER. Obwohl der Copper jedes Bit dieses Registers ändern könnte, wurde dieses speziell für seine Interruptanforderungen reserviert. Es zeigt an, daß die Videoausgabe eine bestimmte Position erreicht hat. Das kann benutzt werden, um bestimmte Parameter in Abhängigkeit von der Videoausgabe zu ändern.

Bit 3: I/O- und Timer-Interrupt Ebene 2

Dieses Signal kommt vom I/O-Baustein 8520-B. Wie schon beim externen Interrupt gibt es auch dort mehrere vormaskierbare Quellen:

- Unterlauf von Timer A
  Keyboard Datenrate

- Unterlauf von Timer B
  Virtual-Timer für die Umschaltung der einzelnen Tasks des Multiuser-Betriebssystems
- TOD-Alarm
  Alarmzeit der Echtzeituhr erreicht
- Serieller Port voll/leer
  Daten vom Keyboard empfangen oder ans Keyboard gesendet
- negative Flanke an $\overline{\text{FLAG}}$ aufgetreten
  Leitung $\overline{\text{ACK}}$ am Parallelport hat von HIGH nach LOW gewechselt

Bit 1: Disk-Block fertig Ebene 1

Zeigt an, daß der angeforderte DMA-Blocktransfer vom Disk-Controller abgeschlossen wurde.

Bit 0: Serieller Port Ebene 1

Das in den Sendepuffer der seriellen Schnittstelle geschriebene Byte wurde in das Schieberegister übernommen. Der Sendepuffer ist damit frei für weitere Daten.

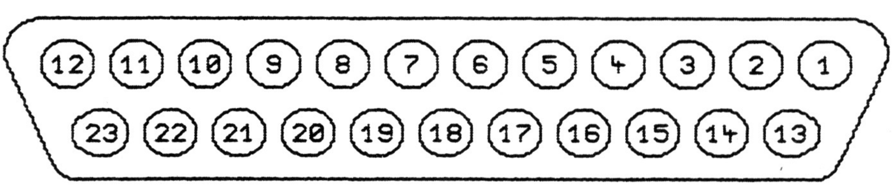
3.6 Disk-Betrieb

Der Disk-Controller im Amiga kann bis zu vier doppelseitige $3^1/_2$- oder $5^1/_4$-Zoll Laufwerke handhaben. Die Steuerung geschieht dabei über einen Bus, an den alle Geräte parallel angeschlossen werden.

3.6.1 Steuerung

Der Spezialbaustein PAULA beherbergt einen Disk-Controller, der im wesentlichen den Datentransfer mittels DMA (Direct Memory Access = direkter Speicherzugriff) über drei Leitungen dieses Busses managt. Für die außerdem nötigen Steuersignale sind Teile der beiden I/O-Bausteine 8520 eingesetzt. Genaue Angaben enthält Tabelle 3.20 mit der Belegung des 23-poligen Disk-Steckers an der Rückseite aller Amiga-Modelle. Gepufferte Signale führen in ihrem Namen ein nachgestelltes B (buffered).

Bild 3.23 zeigt die genaue Beschaltung des internen Laufwerks. Außer dem vom Flipflop generierten Signal $\overline{\text{MTR0D}}$ führen alle Leitungen auch zum Disk-Stecker an der Geräterückseite. Das Select-Signal $\overline{\text{SEL0}}$ erscheint nicht am externen Stecker. Es kommt vom Portbaustein 8520-B, Anschluß PB3 und steuert das eingebaute Laufwerk. Beim Amiga 2000 ist auch das zweite Laufwerk (Nummer 1) intern vorgesehen. Die Ansteuerelektronik ist bereits auf der Hauptplatine enthalten. Daher wurde dort $\overline{\text{SEL1B}}$ ebenfalls nicht herausgeführt. An Pin 21 (sonst $\overline{\text{SEL1B}}$) erhält man $\overline{\text{SEL2B}}$, Pin 9 (sonst $\overline{\text{SEL2B}}$) führt $\overline{\text{SEL3B}}$ und Pin 20 (sonst $\overline{\text{SEL3B}}$) bleibt unbeschaltet. Das erste extern am Amiga 2000 angesteckte Laufwerk erhält damit die Nummer 2, das zweite die Num-



Pin	Signal	Anschluß intern		Funktion
1	$\overline{\mathrm{RDY}}$	8520-A	PA5	Disk Ready
2	$\overline{\mathrm{DKRD}}$	PAULA		Disk Read Data
3	GND	GND		Signalmasse
4	GND	GND		Signalmasse
5	GND	GND		Signalmasse
6	GND	GND		Signalmasse
7	GND	GND		Signalmasse
8	$\overline{\mathrm{MTRXD}}$	8520-B	PB7	Disk-Motor an (wird vom Laufwerk bei Aktivierung des entsprechenden Select-Signals zwischengespeichert)
9	$\overline{\mathrm{SEL2B}}$	8520-B	PB5	Disk-Select Laufwerk 2 (bei Amiga 2000 $\overline{\mathrm{SEL3B}}$)
10	$\overline{\mathrm{DRESB}}$	Reset		Systemreset gepuffert
11	$\overline{\mathrm{CHNGE}}$	8520-A	PA2	HIGH = Disk entnommen (wird bis zum nächsten Step-Impuls gelatcht)
12	+5V	+5 Volt		Versorgungsspannung
13	$\overline{\mathrm{SIDEB}}$	8520-B	PB2	Disk-Kopfauswahl: 0 oben 1 unten
14	$\overline{\mathrm{WPRO}}$	8520-A	PA3	Disk ist schreibgeschützt
15	$\overline{\mathrm{TK0}}$	8520-A	PA4	Disk-Kopf auf Spur 0
16	$\overline{\mathrm{DKWEB}}$	PAULA		Disk Write Enable
17	$\overline{\mathrm{DKWDB}}$	PAULA		Disk Write Data
18	$\overline{\mathrm{STEPB}}$	8520-B	PB0	Stepperschritte für Kopfbewegung Muß nach jedem Puls unbedingt wieder HIGH werden
19	$\overline{\mathrm{DIRB}}$	8520-B	PB1	Step-Richtung für Kopfbewegung 0 zur Diskmitte 1 nach Außen (Spur 0 ist außen)
20	$\overline{\mathrm{SEL3B}}$	8520-B	PB6	Disk-Select Laufwerk 3
21	$\overline{\mathrm{SEL1B}}$	8520-B	PB4	Disk-Select Laufwerk 1 (bei Amiga 2000 $\overline{\mathrm{SEL2B}}$)
22	$\overline{\mathrm{INDEX}}$	8520-B	$\overline{\mathrm{FLAG}}$	Disk-Index-Signal
23	+12V	+12 Volt		Versorgungsspannung

Tabelle 3.20: Die Belegung des Disk-Steckers bei Amiga 500, 1000 und 2000

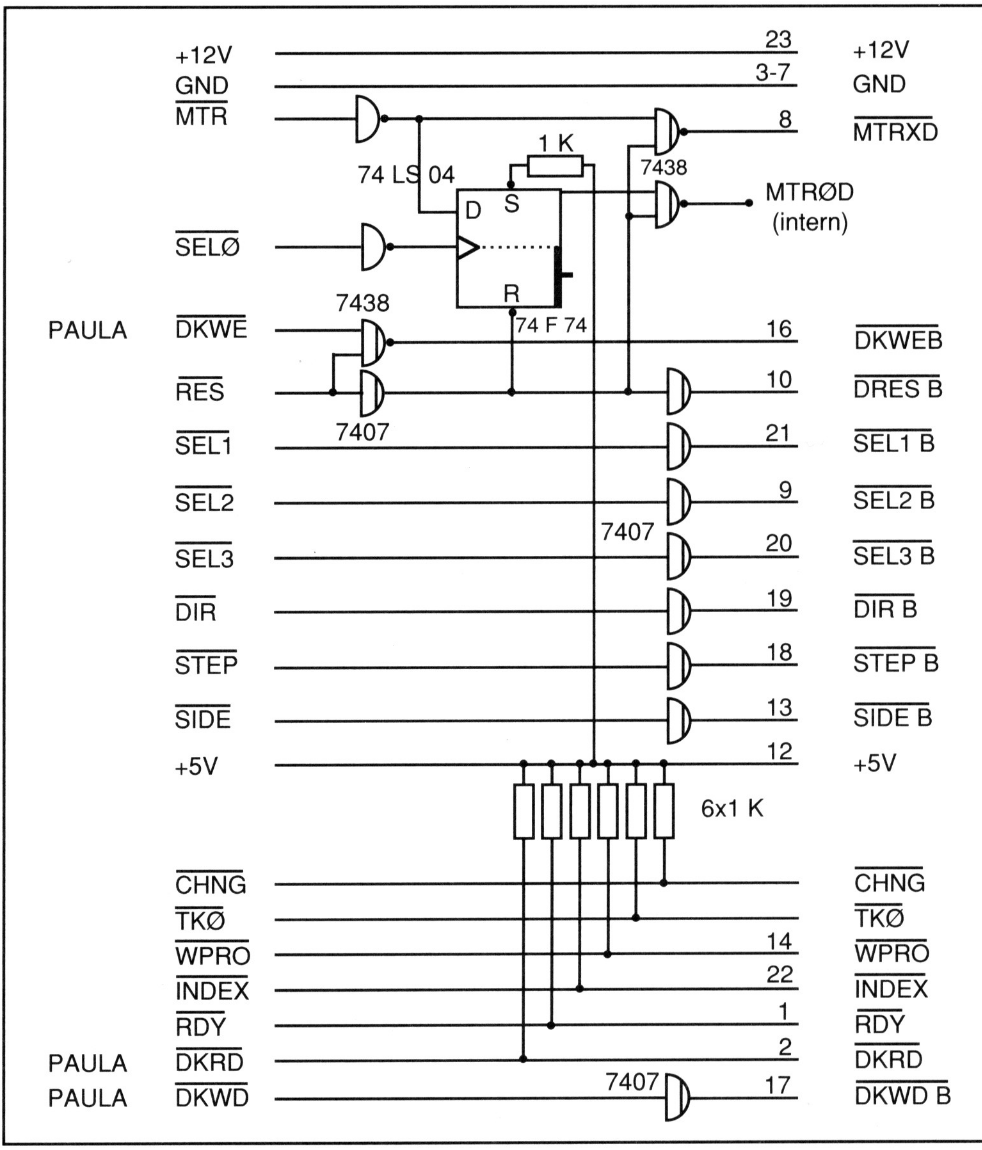


Bild 3.23: Ansteuerung der Laufwerke im Amiga

mer 3 und alle weiteren werden nicht erkannt. Auf ähnliche Weise werden die Select-Leitungen beim Durchschleifen der Anschlüsse zu weiteren externen Laufwerken vertauscht, wie in Bild 3.24 zu sehen. Die Einstellung einer Laufwerksnummer am Gerät wird dadurch überflüssig.

Am Anschluß $\overline{\text{STEP}}$ erhält der Schrittmotor im Laufwerk Impulse zur Kopfpositionierung. Die Zusammenhänge wurden bereits in Kapitel 2.3.5 erläutert. Dort finden Sie auch die Beschreibung des Programms TDChange, mit dem die Step-Parameter geändert werden können.

Damit externe Laufwerke auch längere Zuleitungen haben können, werden alle Ausgangssignale mit Open-Collector-Treibern gepuffert. Die nötigen Pull-up-Widerstände sitzen im Laufwerk selbst. Wie in Kapitel 1 bereits erwähnt, ist diese Verschaltung bei längeren Übertragungsleitungen sinnvoll. Außerdem bedeuten die Treiber einen Schutz gegen Überlastung der 8520-Ports. Beim Anschluß von mehreren Laufwerken sollten die Abschlußwiderstände nur im letzten Gerät beschaltet sein. In den meisten Geräten sind sie daher in IC-ähnlichen Gehäusen untergebracht und können leicht entfernt werden.

3.6.2 Disk - Variation

Wegen der von anderen Geräten etwas abweichenden Motorsteuerung benötigt der Amiga Spezialllaufwerke, die natürlich etwas teurer gehandelt werden als vergleichbare Standardgeräte. Es ist jedoch ohne Schwierigkeiten möglich, nach dem Vorbild der internen Verschaltung mit einem kleinen Zusatz gewöhnliche Laufwerke an den Amiga anzupassen.

Normalerweise werden die Laufwerksmotoren beim Selektieren eines Laufwerks eingeschaltet. Um aber mehrere Laufwerke quasi gleichzeitig betreiben zu können, haben die Entwickler sich eine Speicherschaltung für die Motorsteuerung ausgedacht. Der Zustand der Motorsteuerleitung $\overline{\text{MTRXD}}$ wird bei jedem Laufwerk in dem Moment zwischengespeichert, in dem ein Select-Impuls auf der entsprechenden $\overline{\text{SEL}}$-Leitung erscheint. Zum Einschalten von Laufwerk 0 (dem internen Drive) wird also das LOW-aktive Signal $\overline{\text{MTRXD}}$ durch Löschen von PB7 am 8520-B auf LOW gelegt. Dann folgt ein kurzer Impuls an $\overline{\text{SEL0}}$ und der Motor läuft an. Er dreht die Diskette auch weiter, wenn $\overline{\text{MTRXD}}$ wieder nach HIGH geht und stoppt erst nach einem erneuten Impuls an $\overline{\text{SEL0}}$ unter der Bedingung, daß währenddessen HIGH an $\overline{\text{MTRXD}}$ liegt.

So können nicht nur die Laufwerksmotoren unabhängig voneinander geschaltet werden, sondern es ist auch ein quasi gleichzeitiger Diskettenbetrieb möglich. Nachdem bei einem Laufwerk der Schreib-/Lesekopf auf die gewünschte Spur positioniert wurde, kann während der Wartezeit auf eine Sync-Erkennung beispielsweise ein anderes Laufwerk bereits hochgefahren werden.

Eine kleine Interface-Schaltung zur Anpassung gängiger Laufwerke ist leicht aufzubauen. Bild 3.24 zeigt ihren Schaltplan.

Die Speicherung des Motor-Signals geschieht über ein D-Flipflop 74LS74, wobei $\overline{\text{MTRXD}}$ den D-Anschluß belegt und das invertierte Selectsignal den Taktimpuls bildet. Die Pull-up-Widerstände sind wegen der Schaltung mit Open-Collector-Gattern nötig. Damit nach dem Einschalten des Rechners oder nach einem Reset die Laufwerke zunächst deselektiert sind, wird das D-Flipflop von dem am Floppy-

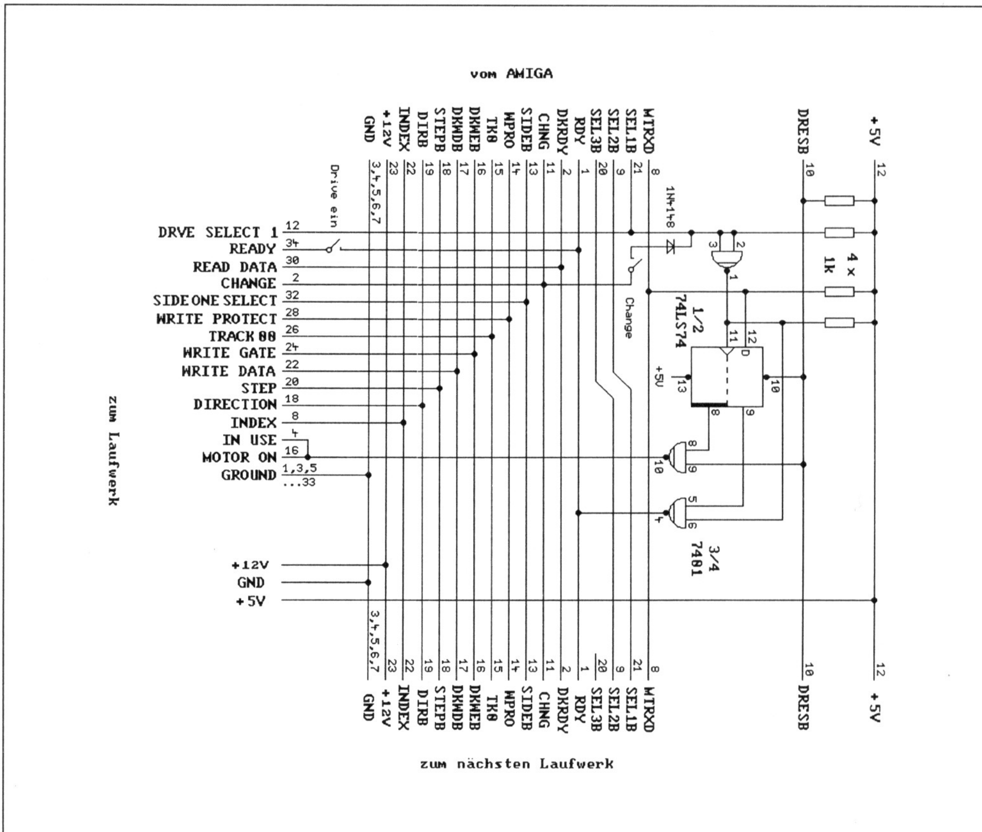


Bild 3.24: Interface zum Anschluß von Standardlaufwerden an den Amiga

Stecker vorhandenen gepufferten Systemresetsignal $\overline{DRESB}$ in eine stabile Anfangslage (Q = HIGH, $\overline{Q}$ = LOW) gebracht.

In Original-Amiga-Laufwerken befindet sich jeweils ein Stecker, an den sich das nächste Laufwerk anschließen läßt. Die Verschaltung der Select-Leitungen geschieht dabei wie in Bild 3.24 dargestellt. $\overline{SEL1B}$ endet im angeschlossenen Laufwerk, $\overline{SEL2B}$ wird als $\overline{SEL1B}$ und $\overline{SEL3B}$ als $\overline{SEL2B}$ ans nächste Laufwerk weitergeleitet, während der Ausgang für

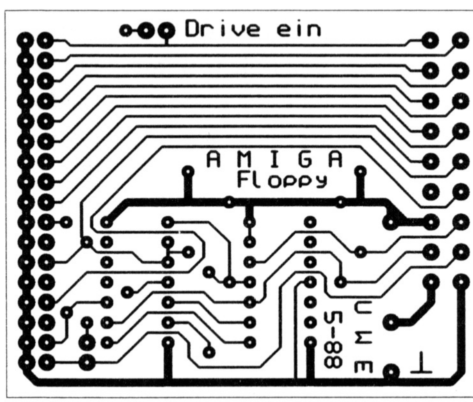


Bild 3.25: Platine für ein Standardlaufwerk am Amiga

$\overline{SEL3B}$ unbeschaltet bleibt. Da jedes angesteckte Laufwerk das ihm an $\overline{SEL1B}$ angebotene Signal benutzt, ist so automatisch die richtige Ansteuerleitung wirksam.

Alle anderen Signalleitungen können bei Bedarf ohne weitere Verschaltung durchgeschleift werden. Allerdings wurde auf der kleinen Platine nach Bild 3.25 ein solcher Stecker nicht vorgesehen. Tabelle 3.21 listet die benötigten Bauteile auf. Richten Sie sich bei der Bestückung nach Bild 3.26 und den Hinweisen in Anhang A.

Mit der Interface-Schaltung stehen alle Signale zum Anschluß der 34-poligen Pfostensteckerleiste eines Standardlaufwerkes zur Verfügung. Tabelle 3.22 zeigt

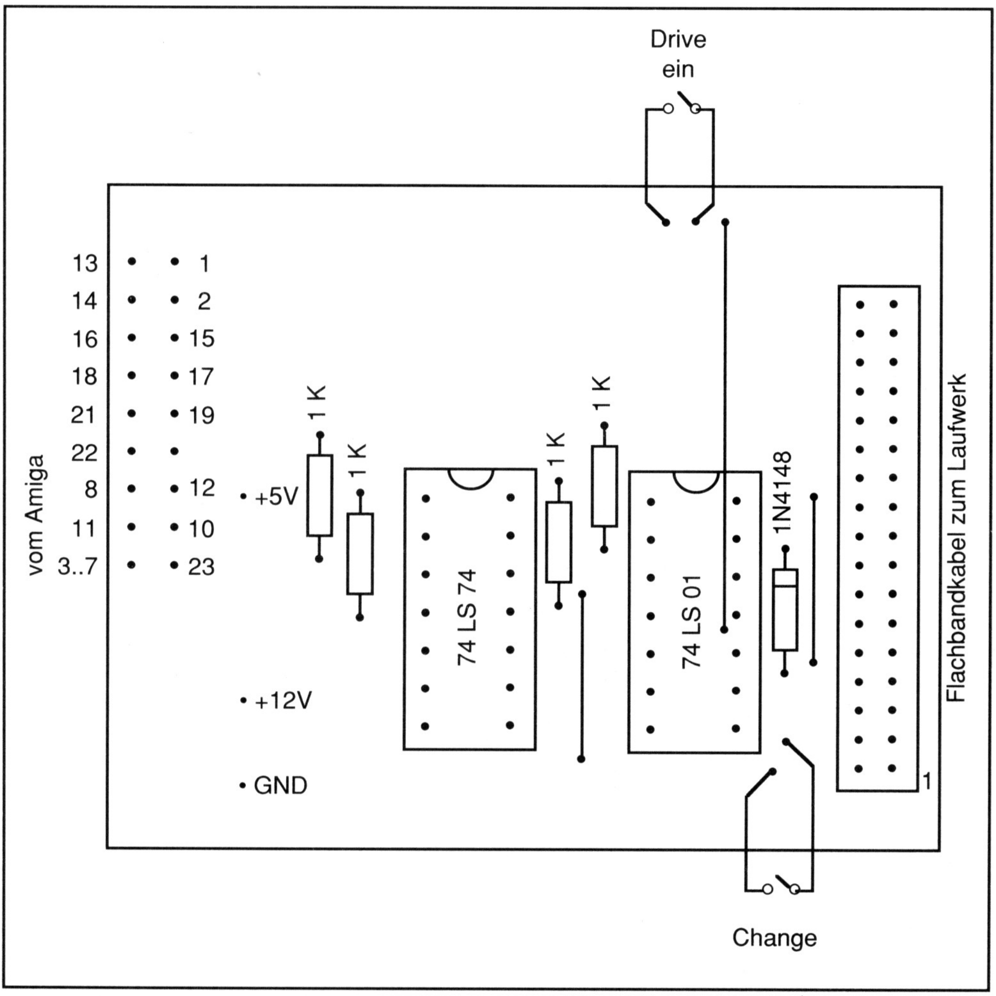


Bild 3.26: Bestückung der Disk-Platine

1	74LS01
1	74LS74
2	IC-Sockel, 14-pol
4	Widerstände 1 Kiloohm
1	Pfostensteckerleiste, 34-pol
1	Pfostenbuchsenleiste, 34-pol, Kontakte quetschbar
1	Floppy-Stecker, 34-pol, Kontakte quetschbar
1m	Flachbandkabel, 34-pol
1	Floppy-Stromversorgungsbuchse, 4-pol
1	Sub-D-Stecker, 23-pol mit Gehäuse
1m	Rundkabel, 17-pol

Tabelle 3.21: Die benötigten Bauteile für ein neues Laufwerk am Amiga

die Zuordnung der Anschlüsse. Alle Pins mit ungeraden Nummern sind mit Masse zu verbinden.

Falls das angeschlossene Laufwerk keine Diskettenwechsel erkennt, muß noch der Schalter S1 eingebaut werden. Er ist vor dem Entnehmen der Disk zu schließen und nach dem Einlegen einer neuen wie-

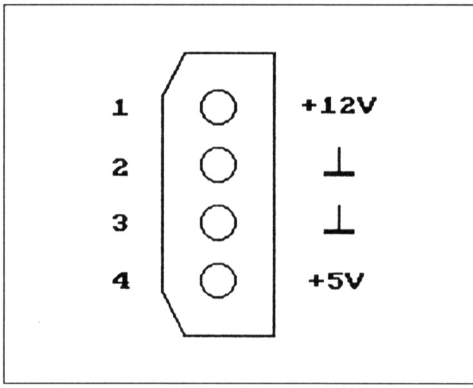


Bild 3.27: Stromversorgungsstecker des Laufwerks in der Draufsicht

der zu öffnen. Eleganter ist natürlich der Einbau eines Mikroschalters ins Laufwerk. Mit etwas Geschick läßt er sich vor allem bei großen 5¹/₄ Zoll-Laufwerken mit dem Schließmechanismus koppeln.

Jedes angeschlossene Laufwerk wird vom Betriebssystem des Amiga selbständig erkannt. Dabei wird ihm unter anderem auch ein Pufferspeicher zugeteilt. Das kann bei manchen Anwendungen unerwünscht sein. Daher läßt sich das Lauf-

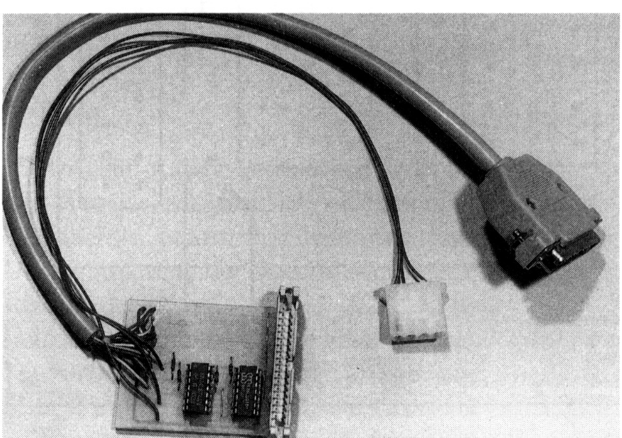


Foto 3.5: Disk-Adapter

Laufwerk Pin	Amiga Pin	Signal
Pfostenstecker		
2	11	$\overline{\text{CHNG}}$
4	Interface	$\overline{\text{in use}}$
6	--	--
8	22	$\overline{\text{INDEX}}$
10	21	$\overline{\text{SEL1B}}$
12	--	--
14	--	--
16	Interface	$\overline{\text{motor on}}$
18	19	$\overline{\text{DIRB}}$
20	18	$\overline{\text{STEPB}}$
22	17	$\overline{\text{DKWDB}}$
24	16	$\overline{\text{DKWEB}}$
26	15	$\overline{\text{TK0}}$
28	14	$\overline{\text{WPRO}}$
30	2	$\overline{\text{DKRD}}$
32	13	$\overline{\text{SIDEB}}$
34	1	$\overline{\text{RDY}}$
1,3,5,7 ... 33	3,4	GND
Stromversorgung		
1	12	+5V
2,3	3,4	GND
4	23	+12V

Tabelle 3.22: Pinzuordnung beim Anschluß eines Standardlaufwerks

werk mit dem Schalter »Drive ein« wahlweise zuschalten. Der Schalter muß vor dem Einschalten oder einem Reset in die gewünschte Stellung gebracht werden.

In Bild 3.27 ist die Belegung des üblichen Stromversorgungssteckers zu sehen. Bei allen Laufwerken muß auf die richtige Stellung der Jumper geachtet werden. Tabelle 3.23 gibt die Sollwerte sowie die Lage der Lötbrücken auf der Platine eines NEC-1035-Laufwerkes vom Schnittstellenstecker aus betrachtet an.

Jumper J1	gesetzt
Jumper DCG	mitte/links (Stellung 1)
Lötbrücke DSX	mitte/links
Lötbrücke DCR	mitte/oben
Jumper M1	gesetzt
Jumper MN	mitte/oben (Stellung 2)
Jumper M0	gesetzt
Lötbrücke M2	offen
Jumper DL	abgezogen
Jumper HL	mitte/oben (Stellung 2)

Tabelle 3.23: Lage der Jumper und Lötbrücken beim Laufwerk NEC 1035

3.7 DMA

Normalerweise muß beim Verschieben von Daten innerhalb des Speichers oder beim Einlesen von einer festen Quelle jedes einzelne Byte zunächst vom Prozessor gelesen und dann unverändert in die vorgesehene Adresse geschrieben werden. Das kostet viel Zeit, denn prozessorintern muß beispielsweise jedesmal die Adressierung softwaremäßig aktualisiert werden.

Es liegt daher nahe, die ohnehin immer gleichen Vorgänge von einer speziellen Hardware ausführen zu lassen, und genau diese Möglichkeit wird vom Amiga in vielen Bereichen intensiv genutzt. Die Daten werden unter Umgehung des Prozessors transferiert, indem ein DMA-Controller direkt auf den Speicher zugreift. DMA steht für Direct Memory Access (direkter Speicherzugriff).

3.7.1 Prozessor abgehängt

Insgesamt verfügt der Amiga über neun DMA-Kanäle, deren Aktivität ähnlich wie bei Interrupts durch ein Register kontrolliert wird. Es handelt sich um DMACON ($DFF096). Unter DMACONR ($DFF002) läßt sich der Zustand dieses Registers auch lesen, so daß leicht festgestellt werden kann, welche DMA-Transfers gerade aktiv sind. Außerdem sind noch einige Blitter-DMA-Statusbits enthalten. Tabelle 3.24 enthält Einzelheiten.

Wie bei den Interrupt-Registern können während eines Zugriffs die Bits nur entweder gesetzt oder gelöscht werden.

Bei einem DMA-Transfer wird der Prozessor zumindest zeitweise angehalten. Er gibt den internen Bus frei und AGNUS erzeugt die Adreßpegel, während der Chip, in dem die aktive DMA-Logik sitzt, je nach Programmierung direkt aus dem RAM Daten liest oder in das RAM Daten schreibt. Der Geschwindigkeitsgewinn ist offensichtlich.

Zur Adressierung des RAMs besitzt AGNUS neun Bausteinanschlüsse, was bedeutet, daß DMA-Zugriffe nicht im gesamten Speicher ausgeführt werden können. Da es sich bereits um gemultiplexte Signale handelt, lassen sich 2 hoch 18, also 262144 Adreßkombinationen erzeugen, entsprechend 256K Worte oder 512 Kilobyte. Die unteren 512 Kilobyte Speicher werden wegen dieser Zugriffsmöglichkeit durch AGNUS »Chip-RAM« genannt, im Unterschied zum sonstigen sogenannten »Fast-RAM«.

Der im Amiga 500 und im Amiga B 2000 eingesetzte, um einige Funktionen erweiterte Baustein FAT AGNUS wurde um eine Adreßleitung erweitert. Sie dient jedoch ausschließlich zum Refresh eines größeren RAM-Bereiches und ist für DMA-Anwendungen leider nicht verwendbar. Erst im geplanten Amiga 2000 UX sollen wesentlich verbesserte Spezialchips Verwendung finden, die unter anderem flimmerfreie hochauflösende Grafikausgabe erlauben.

Manchmal wird kritisiert, daß der mit knapp über 7 MHz ohnehin eher langsam getaktete 68000-Prozessor im Amiga zugunsten der anderen Bausteine auch noch zeitweise angehalten wird. Das hat jedoch seinen Sinn. Der DMA-Controller in AGNUS kann reine Speicherzugriffe wie

Bit	Funktion	Bedeutung
15	SET/ CLEAR	Legt fest, ob gesetzte Bits die entsprechenden Stellen löschen oder setzen. (1 = setzen, 0 = löschen)
14	BBUSY	Blitter aktiv (nur lesen)
13	BZERO	Blitter Zero Status (nur lesen)
12,11		(unbenutzt)
10	BLTPRI	Blitter hat volle Priorität über den 68000
9	DMAEN	Generelle DMA Freigabe für alle Kanäle
8	BPLEN	Bit-Plane DMA freigegeben
7	COPEN	Copper-DMA freigegeben
6	BLTEN	Blitter-DMA freigegeben
5	SPREN	Sprite-DMA freigegeben
4	DSKEN	Disk-DMA freigegeben
3	AUD3EN	DMA Audiokanal 3 freigegeben
2	AUD2EN	DMA Audiokanal 2 freigegeben
1	AUD1EN	DMA Audiokanal 1 freigegeben
0	AUD0EN	DMA Audiokanal 0 freigegeben

Tabelle 3.24: Die Bits in den DMA-Kontrollregistern

gesagt wesentlich schneller aufeinander folgend ausführen als der Prozessor. Dazu existiert intern spezielle Hardware, wie zum Beispiel ein Adreßzähler. Der Blitter setzt auf diese Weise mittels DMA pro Sekunde etwa eine Million Punkte. Das heißt, Flächen werden nicht mehr ausgefüllt, sie sind einfach mit einem Schlag farbig.

Gerade durch diese Aufgabenteilung wird das Gesamtsystem schneller als ein vergleichbares mit höherem Takt und ohne »Wartezeiten«, bei dem der Prozessor alles selbst erledigen muß.

3.7.2 Disk-DMA

Hier soll die DMA-Programmierung anhand der Diskettenübertragung erläutert werden. Vor jedem Zugriff sind dazu vier Angaben zu machen:

- Anfang des Datenbereichs, von dem oder in den Daten übertragen werden sollen,
- Länge des zu übertragenden Datenblocks,
- Richtung der Datenübertragung und
- DMA-Freigabe.

Bit	Name	Bedeutung
0-13	LENGTH	Anzahl der zu übertragenden Worte
14	WRITE	Richtung: 0 = zum Computer 1 = zur Disk
15	DMAEN	1 = Disk-DMA freigegeben

Tabelle 3.25: Bedeutung der Bits im Register DSKLEN

Zuerst kommt die 19 Bit lange Startadresse in die Register DSKPTH ($DFF020) und DSKPTL ($DFF022). Dabei erhält DSKPTL die niederwertigen 16 Bit, DSKPTH die höchstwertigen 3. Bit 0 muß immer gelöscht sein, da Transfers nur auf geraden Adressen starten können.

Länge, Richtung und DMA-Freigabe stehen gemeinsam im Register DSKLEN ($DFF024). Tabelle 3.25 erläutert die Bedeutungen der einzelnen Bits. Alle drei genannten Register können nur beschrieben werden.

Um versehentliches Schreiben auf die Diskette zu vermeiden, ist vor einer Übertragungssequenz jeweils das Freigabe-Bit zu löschen. Dies geschieht durch Schreiben von $4000 nach DSKLEN. Dann setzen Sie die oben beschriebenen Parameter und beschreiben das DSKLEN-Register zweimal. Erst dann startet die Übertragung. Nach deren Abschluß ist es sinnvoll, DSKLEN wieder auf $4000 zurückzustellen, um nachfolgend unbeabsichtigte Diskettenzugriffe zu vermeiden.

An dieser Stelle soll noch auf einen Fehler in der Hardware zumindest älterer Geräte hingewiesen werden. Beim Schreiben auf Diskette gehen die letzten drei Bit verloren. Beim Lesen kann es vorkommen, daß das letzte Wort nicht übertragen wird. Daher ist immer ein Wort mehr anzufordern, als eigentlich nötig.

3.7.3 Audio-DMA

In Kapitel 2.7 wurde ein Digitizer vorgestellt. Die gesampelten Daten kann der Amiga über vier Audio-Kanäle ausgeben. Dabei geschieht die Datenübertragung auch hier wieder mittels DMA. Das DMA-Kontrollregister haben Sie bereits in Tabelle 3.24 kennengelernt. Dort lassen sich auch die vier Audio-Kanäle aktivieren. Die Vorgänge ähneln denen beim Disk-DMA. Auch hier wird in einem Register die Adresse übergeben, bei der die Daten im Speicher beginnen, in einem anderen, wie groß der Bereich ist. Daneben ist noch die Ausgabegeschwindigkeit sowie die Lautstärke festzulegen.

Die Programmierung der vier Audio-Kanäle ist grundsätzlich gleich. Für jeden Kanal gibt es die genannten Register. Sie unterscheiden sich im Namen nur durch die Kanalnummer. In Tabelle 3.26 sind die Register, ihre Bedeutung und ihre Adressen zusammengestellt.

Alle Register sind nur für Schreibzugriffe vorgesehen. Im folgenden soll für die Ka-

AUD0LCH	$DFF0A0	Audiokanal 0 Startadresse (Bits 16..18)
AUD0LCL	$DFF0A2	Audiokanal 0 Startadresse (Bits 1..15)
AUD0LEN	$DFF0A4	Audiokanal 0 Länge in Worten
AUD0PER	$DFF0A6	Audiokanal 0 Periode
AUD0VOL	$DFF0A8	Audiokanal 0 Lautstärke
AUD0DAT	$DFF0AA	Audiokanal 0 Daten (nur für DMA-Controller)
AUD1LCH	$DFF0B0	Audiokanal 1 Startadresse (Bits 16..18)
AUD1LCL	$DFF0B2	Audiokanal 1 Startadresse (Bits 1..15)
AUD1LEN	$DFF0B4	Audiokanal 1 Länge in Worten
AUD1PER	$DFF0B6	Audiokanal 1 Periode
AUD1VOL	$DFF0B8	Audiokanal 1 Lautstärke
AUD1DAT	$DFF0BA	Audiokanal 1 Daten (nur für DMA-Controller)
AUD2LCH	$DFF0C0	Audiokanal 1 Startadresse (Bits 16..18)
AUD2LCL	$DFF0C2	Audiokanal 1 Startadresse (Bits 1..15)
AUD2LEN	$DFF0C4	Audiokanal 1 Länge in Worten
AUD2PER	$DFF0C6	Audiokanal 1 Periode
AUD2VOL	$DFF0C8	Audiokanal 1 Lautstärke
AUD2DAT	$DFF0CA	Audiokanal 1 Daten (nur für DMA-Controller)
AUD3LCH	$DFF0D0	Audiokanal 1 Startadresse (Bits 16..18)
AUD3LCL	$DFF0D2	Audiokanal 1 Startadresse (Bits 1..15)
AUD3LEN	$DFF0D4	Audiokanal 1 Datenbereichlänge in Worten
AUD3PER	$DFF0D6	Audiokanal 1 Periode
AUD3VOL	$DFF0D8	Audiokanal 1 Lautstärke
AUD3DAT	$DFF0DA	Audiokanal 1 Daten (nur für DMA-Controller)

Tabelle 3.26: Die Audio-Register

nalnummer jeweils ein x eingesetzt werden. AUDxDAT steht demnach für AUD0-DAT, AUD1DAT, AUD2DAT oder AUD3-DAT.

Die Audio-Daten müssen für DMA-Betrieb im »Chip-Ram« stehen. Ihre Anfangsadresse wird dem System in AUDXLC mitgeteilt. Dabei kann einfach ein Langwortzugriff erfolgen.

Für AUDxPER stehen volle 16 Bit zur Verfügung. Allerdings sollte der Wert mindestens 124 betragen, da andernfalls nicht mehr genügend Zeit zur Ausführung der DMA-Zugriffe bleibt. Die zulässigen Werte für AUDxVOL liegen zwischen 0 (Kanal stumm) und 64 (maximale Lautstärke).

Die Soundausgabe wird über das Register DMACONW durch Setzen von Bit 9 gestartet. Gleichzeitig ist der gewünschte Audiokanal mit Bit 0 bis 3 zu aktivieren. Beispielsweise aktiviert die Kombination

```
15            8 7            0

1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1
```

die Audiokanäle 0 und 1. Durch das gesetzte Bit 15 werden die den übrigen gesetzten Bits entsprechenden Funktionen aktiviert.

Die Bitkombination

```
15             8 7             0

0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0
```

dagegen würde den laufenden Audio-DMA auf Kanal 1 wieder abschalten.

Während ein DMA-Zyklus läuft, kann sich die CPU anderen Aufgaben widmen. Erst nachdem der aktuelle Block vollständig ausgegeben wurde, meldet sich die Hardware mit einem Interrupt zurück. Genauere Informationen dazu enhält Kapitel 3.5. Da auch mehrere DMA-Kanäle zur gleichen Zeit aktiv sein können, ist es beispielsweise möglich, während einer laufenden Sound-Ausgabe bereits den nächsten Datenblock von der Diskette an eine andere Stelle des Speichers zu laden, und in der Interruptbehandlung die Ausgabe dort fortzusetzen. Der alte Block kann dann wieder mit den nächsten Daten überschrieben werden. So ist eine lückenlose Soundausgabe von Diskette oder gar Harddisk zu realisieren.

4

Der Video-Star

Wie erzeugt der Amiga sein Videobild? Warum kann er genau 4096 Farben darstellen? Was ist der Unterschied zwischen Digital- und Analog-RGB? Was steckt hinter (F)BAS? Welche Auflösung hat der Amiga wirklich? Was bedeutet eigentlich Interlace? Wie arbeitet ein Genlock-Interface? Wie funktioniert ein Lightpen?

Antwort unter anderem auf diese Fragen gibt das folgende Kapitel.

4.1 Bilderzeugung

Die meisten Computermonitore arbeiten nach der Fernsehnorm. Sehen wir uns daher kurz an, wie das Bild sowohl beim Fernseher, als auch beim üblichen Monitor auf der Mattscheibe entsteht, und was die digitale Bilderzeugung bewirkt.

4.1.1 Fernsehen stand Pate

Die Entwickler des Fernsehens wollten bewegte Bilder übertragen. Ähnlich wie

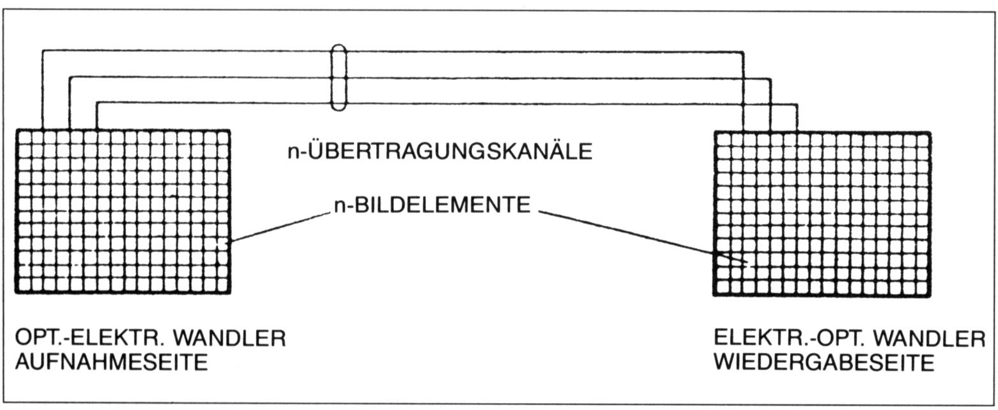


Bild 4.1: Bildübertragung mit Vielfachverbindung

beim Film gingen sie dabei von der Idee aus, dem Betrachter viele einzelne Momentaufnahmen kurz hintereinander zu präsentieren. Im Gegensatz zu auf eine Leinwand projezierten Dias kann aber elektronisch ein gesamtes Bild nur mit großem Aufwand erzeugt werden. Man setzte es daher aus einzelnen Bildpunkten wie ein Mosaik zusammen. Grob gesagt könnte man für jedes Mosaikteil eine Glühlampe einbauen, die mehr oder weniger hell leuchtet. Für jede Glühlampe wäre dann aber mindestens ein Steuerkanal nötig. Bild 4.1 läßt ahnen, daß eine solche Übertragung viel zu aufwendig wäre.

Doch schon bei der Auflösung einer fließenden Bewegung in Einzelbilder beim Film läßt sich ja das Auge täuschen. Diese Schwäche wird konsequent ausgenutzt, indem man auch bei jedem Bild die einzelnen Punkte nur kurz, dafür aber schnell aufeinander folgend zeigt. Dabei wird die Vorlage nach Bild 4.2 zeilenweise von links oben nach rechts unten abgetastet. Natürlich muß dieser Vorgang sehr schnell ablaufen, damit für den Betrachter ein einheitliches Bild entsteht. Aus historischen Gründen wurde die Bildwechselfrequenz auf den Wert der Netzfrequenz festgelegt. Daher arbeitet das Fernsehsystem in Europa mit 50Hz, das in Amerika dagegen mit 60Hz. Bei uns werden also 50 Bilder pro Sekunde erzeugt.

Um bei der Bilderzeugung mit vertretbarem Aufwand auszukommen, arbeitet die Bildröhre mit einem Elektronenstrahl, der

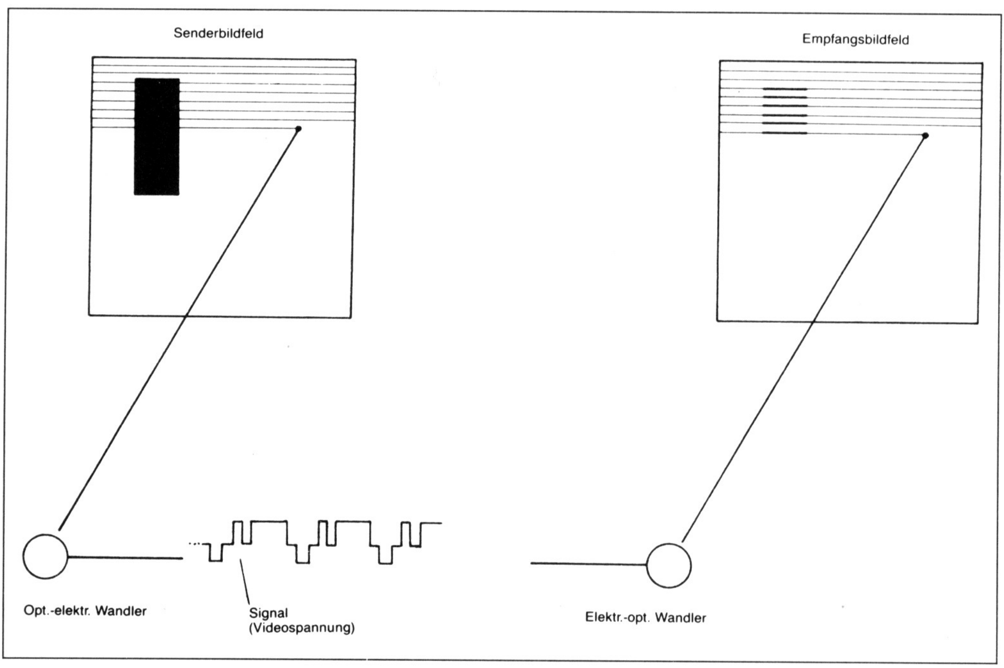


Bild 4.2: Fernsehübertragung mit wanderndem Lichtpunkt

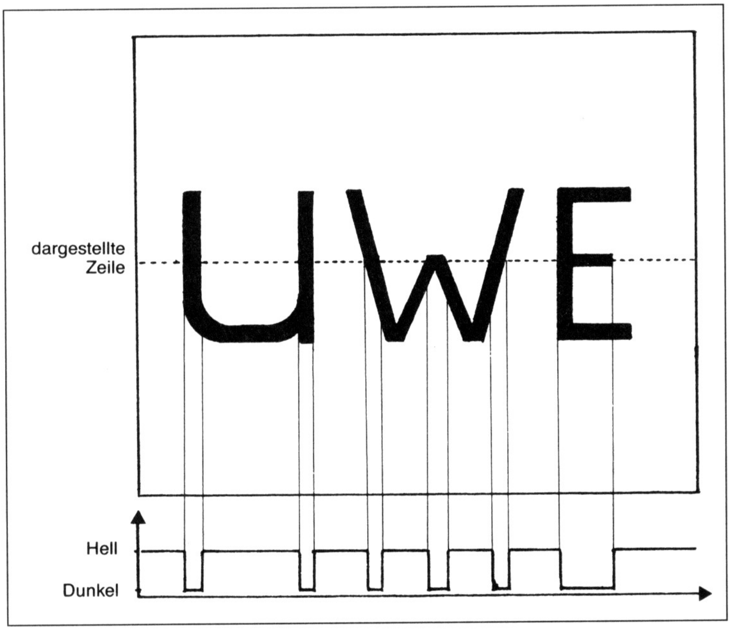


Bild 4.3: Entstehung des Helligkeitssignals

mittels Magnetfeldern auf jeden Punkt der Mattscheibe abgelenkt werden kann und dort einen Lichtpunkt hervorruft. Weiterhin ist die Helligkeit dieses Punktes veränderbar. Sie ist abhängig von der augenblicklichen Videospannung. Bild 4.3 zeigt die Entstehung dieses Spannungsverlaufes für die gestrichelte Zeile. Hell entspricht einem hohen Spannungswert, dem sogenannten Weißwert, und Dunkel einem niedrigen. Spezielle »Synchronimpulse« am Anfang jedes Bildes und jeder Zeile besitzen noch kleinere Spannungen als dieser Schwarzwert, damit der Lichtpunkt, während er zum Anfang der nächsten Zeile zurückwandert, auch tatsächlich dunkel bleibt. Im Fachjargon heißt dieser Vorgang Austastung.

Weil durch die Fernsehnorm genau festgelegt ist, in welcher Reihenfolge und Geschwindigkeit die Zeilen übertragen werden, kann der Empfänger die Positionierung des Lichtpunktes auf der Mattscheibe selbst übernehmen. Das Videosignal besteht damit aus drei Komponeneten: dem Bild-, dem Austast- und dem Synchronsignal. Man spricht daher auch vom BAS-Signal. In Bild 4.2 ist der zusammengesetzte Spannungsverlauf für drei aufeinanderfolgende Zeilen zu erkennen. Um bei den gegebenen Umständen ein hochauflösendes und trotzdem möglichst flackerfreies Bild entstehen zu lassen, arbeitet das Fernsehen mit dem sogenannten Interlace-Verfahren. Dabei wird zunächst nur jede zweite Zeile des vollständigen Bildes dargestellt, nämlich das erste Halbbild, danach die übrigen Zeilen. Ein volles Bild benötigt also zwei Bildaufbauphasen, womit die Vollbild-

frequenz 25Hz beträgt. Ein volles Fernsehbild besteht bei uns aus 625 Zeilen.

Nur wenige Computer können echte Interlace-Bilder erzeugen. Sie stellen gewöhnlich während beider Halbbild-Phasen dieselben Informationen dar. Der Amiga ist für Interlace-Betrieb ausgerüstet. Um diese Betriebsart einzuschalten, muß Bit 2 des Bit Plane-Kontrollregisters 0 (BPLCON0) in Adresse $DFF100 gesetzt werden.

4.1.2 Jetzt kommt Farbe ins Bild

Sie wissen sicherlich, daß jeder Farbton durch Mischen von drei Grundfarben erzeugt werden kann. Diesen Effekt macht man sich auch bei der Übertragung farbiger Bilder zunutze. In der Kamera wird das Bild durch Filter in drei Farbkomponenten, nämlich Rot, Grün und Blau zerlegt. Jede Komponente liefert ganz analog zum Schwarzweißbild ein Helligkeitssignal. Im Empfänger steht für jede Farbe wieder eine Kathodenstrahlröhre zur Verfügung. Durch die Überlagerung der Farben auf der Mattscheibe entsteht das farbige Bild.

Um das Ergebnis über einen einzigen Kanal senden zu können, werden die drei Signale mit Hilfe einer besonderen Modulationstechnik zu einem einzigen Spannungsverlauf zusammengefügt. Es entsteht das Farbart-, Bild-, Austast- und Synchronsignal (FBAS).

Beim Senden eines Fernsehbildes wird das erzeugte Videosignal zusammen mit den Toninformationen in eine hochfrequente Wechselspannung umgewandelt. Man sagt auch, es wird einer Trägerfrequenz aufmoduliert. Im Fernsehempfänger geschieht genau das Gegenteil: Das Gemisch wird demoduliert; das heißt, man erhält wieder das ursprüngliche Videosignal zurück.

In Heimcomputern verfährt man nicht anders. Der Modulator ist für sich genommen ein kleiner Fernsehsender mit geringer Leistung, der direkt über das Antennenkabel an den Empfänger geschaltet wird. Am Videosignal geht diese Prozedur allerdings nicht spurlos vorüber. Wer schon einmal mit Hochfrequenz experimentiert hat weiß, daß entsprechende Schaltungen bereits die kleinste Ungenauigkeit sehr übel nehmen. Um scharfe Zeichen zu erhalten, müssen besonders bei Schriftdarstellung abrupte Übergänge zwischen hell und dunkel unbedingt sauber übertagen werden (Bild 4.3). Die vielen durchlaufenen Verarbeitungsstufen erzeugen aber zwangsweise nach und nach verwaschene Übergänge: Die Zeichen werden unscharf. Besonders bei voller 80-Zeichen-Darstellung ist ein Fernseher in aller Regel unbrauchbar. Die Bilddarstellung läßt sich erheblich verbessern, indem der Umweg über die Hochfrequenz vermieden wird. Daher ist ein Monitor dem Fernseher vorzuziehen.

Am saubersten läuft die Übertragung, wenn gar nicht erst verschiedene Signale zusammengefaßt werden. Das setzt mehrere Leitungen zwischen Sender und Empfänger voraus, was aber bei dicht nebeneinander aufgestellten Einheiten nicht weiter schlimm ist. Im einzelnen sind das neben Masse jeweils eine Verbindung für das Rot-, das Grün- und das

Blausignal, mindestens eine für die Synchronimpulse (Bild- und Zeile) sowie eine bzw. bei Stereo zwei für Ton.

4.1.3 Bit-Show

Bild 4.3 zeigt, daß zur Erzeugung eines Schwarzweißbildes eine Ausgangsleitung genügt, die zur richtigen Zeit ein- oder ausgeschaltet wird. Der Videostrahl im Bildschirm ist dann hell oder dunkel.

Wird ein TTL-RGB-Farbmonitor benutzt, stehen drei Eingänge zur Verfügung. Hohe Spannung nur an R erzeugt ein rotes Bild, Spannung nur an G ein grünes und nur an B ein blaues. Daneben gibt es Mischungen. Tabelle 4.1 listet die darstellbaren Farben auf.

Um mehr Farbmöglichkeiten zu bekommen, bieten manche TTL-Monitore noch einen Intensitätseingang an. Er kontrolliert die Helligkeit des Bildes. Damit ergibt sich die doppelte Zahl darstellbarer Farben.

Das Bild wird jedoch auch hier immer digital erstellt, da jeder der Eingänge nur entweder ein oder aus sein kann.

Rot	Grün	Blau	Farbe
aus	aus	aus	schwarz
aus	aus	ein	blau
aus	ein	aus	grün
aus	ein	ein	zyanblau
ein	aus	aus	rot
ein	aus	ein	magenta
ein	ein	aus	braun
ein	ein	ein	weiß

Tabelle 4.1: Mögliche Farben bei digital RGB

Int.	Rot	Grün	Blau	Farbe
aus	aus	aus	aus	schwarz
aus	aus	aus	ein	blau
aus	aus	ein	aus	grün
aus	aus	ein	ein	zyan-blau
aus	ein	aus	aus	rot
aus	ein	aus	ein	magenta-rot
aus	ein	ein	aus	braun
aus	ein	ein	ein	hellgrau
ein	aus	aus	aus	dunkelgrau
ein	aus	aus	ein	hellblau
ein	aus	ein	aus	hellgrün
ein	aus	ein	ein	hellzyan-blau
ein	ein	aus	aus	hellrot
ein	ein	aus	ein	hellmagenta-rot
ein	ein	ein	aus	(hell)gelb
ein	ein	ein	ein	weiß

Tabelle 4.2: Mögliche Farben bei digital RGB mit Intensity-Signal

Bei Analog-RGB-Monitoren läßt sich dagegen die Helligkeit jedes Kathodenstrahls im Monitor durch Variation der Eingangsspannung stufenlos verändern, so daß tatsächlich wesentlich mehr Farbtöne erzielbar sind. Allerdings muß der Computer dazu über einen Wandler zunächst die entsprechenden Zwischenspannungen erzeugen.

4.1.4 Flotte Analog/Digital-Wandler

Es wurde bereits erwähnt, daß die Funktionen der drei Spezialchips im Amiga eng miteinander verzahnt sind, und daß nicht etwa jeder einen abgeschlossenen Aufgabenbereich versorgt. Ein Beispiel dafür ist die Videobild-Erzeugung. Während die verschiedenen Synchronsignale an AGNUS abgreifbar sind, entsteht der eigentliche Bildinhalt im hochintegrierten Inneren von DENISE. Er liegt dort in noch digitaler Form an jeweils vier Ausgängen für Rot (R0 – R3), Grün (G0 – G3) und Blau (B0 – B3) an. Somit lassen sich pro Farbkanal $2^4 = 16$ Spannungen erzeugen. Dazu wird ein einfacher Analog/Digital-Wandler eingesetzt (Bild 4.4).

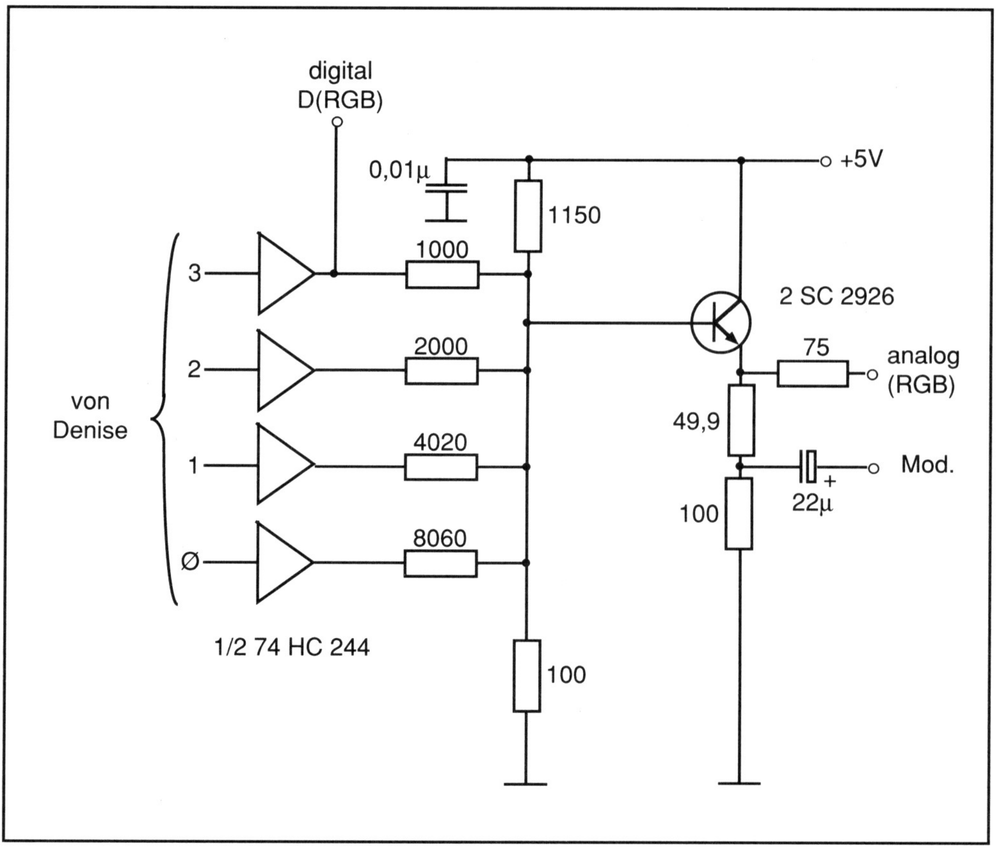


Bild 4.4: Einer der Video-Analog/Digital-Wandler im Amiga

Pin	Signal	Funktion	intern	Richtg.
1	$\overline{XCLK}$	Extern Clock	Taktversorgung	aus
2	$\overline{XCLKEN}$	Extern Clock Enable	Taktversorgung	ein/aus
3	R	analog Rot	Video-Wandler	aus
4	G	analog Grün	Video-Wandler	aus
5	B	analog Blau	Video-Wandler	aus
6	DI	digitale Intensität	DENISE	aus
7	DB	digital Blau	DENISE	aus
8	DG	digital Grün	DENISE	aus
9	DR	digital Rot	DENISE	aus
10	$\overline{CSY}$	Composite-Sync	AGNUS	ein/aus
11	$\overline{HSY}$	Horizontal-Sync	AGNUS	ein/aus
12	$\overline{VSY}$	Vertikal-Sync	AGNUS	ein/aus
13	GND	Masse für $\overline{XCLK}$	Digital-Masse	
14	$\overline{ZD}$	Hintergrundfarbe	DENISE	aus
15	$\overline{C1}$	Taktsignal	Taktversorgung	aus
16	GND	Masse	Video-Masse	
17	GND	Masse	Video-Masse	
18	GND	Masse	Video-Masse	
19	GND	Masse	Video-Masse	
20	GND	Masse	Video-Masse	
21	–5V	–5 Volt	Netzteil	
22	+12V	+12 Volt	Netzteil	
23	+5V	+5 Volt	Netzteil	

Tabelle 4.3: Die Belegung des RGB-Steckverbinders

Jede der vier Digitalstellen hat eine bestimmte Wertigkeit, nämlich genau eins mehr als die Summe aller Wertigkeiten darunter. Die Schaltung soll den digital dargestellten Wert in eine Spannung umsetzen. Jede Bitkombination entspricht so einer bestimmten Ausgangsspannung.

Zunächst werden die digitalen Videospannungen über Treiber vom Typ 74HC244 geführt. An ihren Ausgängen entstehen wegen des sehr niederohmigen Ausgangswiderstandes der CMOS-Innenschaltung einwandfreie Ausgangspegel.

Der nachgeschaltete Transistor wirkt als Summierer. Über genau passende Widerstände ruft jeder Eingang an der Transistor-Basis eine seiner Wertigkeit entsprechende Spannung hervor. Weil drei der gezeichneten Stufen mit jeweils 16 möglichen Spannungen vorhanden sind, ergeben sich 16^3 = 4096 mögliche Farbkombinationen.

Von den Wandlern her gelangen die erzeugten »Analogspannungen« auf den RGB-Steckverbinder (Anschlüsse R, G, B) und von dort zum Monitor. Für RGB-Monitore mit digitalen Eingangsstufen wurden außerdem digitale RGB-Ausgänge vorgesehen (DR, DG, DB).

Die Signale dafür kommen jeweils vom höchstwertigen Bit der drei Farben über einen Widerstand von 47 Ohm. Das niederwertigste Bit der Stufe Blau bildet für diesen Fall auf die gleiche Weise das Intensitätssignal (DI).

An Synchronsignalen stehen sowohl das zeilenfrequente $\overline{\text{HSY}}$, als auch das Bildsynchronsignal $\overline{\text{VSY}}$ zur Verfügung.

Außerdem liegt an $\overline{\text{CSY}}$ das zusammengesetzte Composite-Synchronsignal. Die vollständige Belegung des RGB-Steckers zeigt Tabelle 4.3. Sie ist bei allen Amiga-Modellen gleich.

4.2 Punkt an Punkt

Fast in jeder Veröffentlichung findet man unterschiedliche Angaben zur Grafikauflösung des Amiga.

Welche Daten sind nun richtig?

Eine genaue Antwort auf diese Frage läßt sich prinzipiell nicht geben. Durch ein erfreulich universell angelegtes Ausgabeverfahren ist es nämlich auch möglich, höhere Auflösungen zu fahren, als die von der amerikanischen Fernsehnorm geprägten 320 bzw. 640 mal 200 bzw. 400 Punkte.

Zur Erläuterung der speziellen Eigenschaften soll nun etwas weiter ausgeholt werden.

4.2.1 Völlig aufgelöst

Fangen wir bei AGNUS an. Dort erzeugen zwei gekoppelte Generatoren als Grundlage für die Bilddarstellung laufend die Synchronsignale. Die Vertikalsynchronleitung $\overline{\text{VSY}}$ teilt durch einen kurzen LOW-Impuls mit, daß nun die erste Bildzeile folgt. Der Ablenkteil im Monitor bringt daraufhin sofort seinen Kathodenstrahl in die Ausgangslage am linken oberen Bildschirmrand.

Jede Zeile wird nun in der gleichen Weise mit einem Horizontalsynchronimpuls auf $\overline{\text{HSY}}$ eingeleitet, damit der Monitor den Lichtpunkt rechtzeitig zum linken Bildschirmrand bringt.

In AGNUS laufen ständig zwei softwaremäßig lesbare Zähler mit, die immer die genauen Koordinaten des aktuellen Punktes auf dem Schirm enthalten.

Die Bilddarstellung basiert auf diesem universellen Zeitraster, indem ein Basis-Bildfenster (Display-Window) durch zwei diagonal gegenüberliegende Eckkoordinaten festgelegt wird, nämlich durch einen Startpunkt links oben und einen Stoppunkt rechts unten. Damit ist nicht nur die Lage des nutzbaren Bildschirmteils angegeben, sondern gleichzeitig auch seine Höhe und Breite.

Außerhalb des Basis-Bildfensters wird lediglich die eingestellte Hintergrundfarbe angezeigt. Ab der vorgegebenen Vertikalposition beginnt DENISE jedoch, in jeder Zeile die zugehörigen Informationen auszugeben, so lange, bis die jeweilige Stopposition erreicht ist.

Richtung	Register	Teil	mögliche Werte	Punkte
horizontal	DIWSTRT DIWSTOP	HSTART HSTOP	\$ 00 - \$ FF \$100 - \$1FF	\$200 = 512
vertikal	DIWSTRT DIWSTOP	VSTART VSTOP	\$ 00 - \$ FF \$080 - \$17F	\$100 = 384

Tabelle 4.4: Bedeutung der Register DIWSTRT (\$DFF08E) und DIWSTOP (\$DFF090)

Das Register DIWSTRT (Display Window Start) enthält den linken oberen Eckpunkt, DIWSTOP den rechten unteren. Tabelle 4.4 gibt nähere Informationen.

Wie Sie erkennen, wird jeweils Vertikal- und Horizontalkomponente gemeinsam in einem Register abgelegt. Die Aufteilung ist dabei immer

```
Bit   15 . . . 8    7 . . . 0
       vertikal     horizontal
```

Mit acht Bit lassen sich Werte von \$00 bis \$FF (0 bis 255) darstellen. Damit auch höhere Endwerte festlegbar sind, werden Tricks benutzt. Bei der horizontalen Stopposition ist das (interne) Bit 8 immer gesetzt. Der Wert \$C1 wird also wie \$1C1 behandelt, oder anders ausgedrückt, zum im Register angegebenen Wert addiert der Amiga automatisch 256. Es sind also theoretisch bereits bei niedriger Auflösung maximal 512 Punkte pro Zeile möglich.

Anders bei der vertikalen Stopposition. Dort wird das interne Bit 8 immer durch Invertieren von Bit 7 erzeugt. Die kleinste darstellbare Endzahl ist damit

```
dual          0  1 0 0 0   0 0 0 0
also
hexadezimal   0     8         0
oder dezimal           128
```

die Größte entsprechend

```
dual          1  0 1 1 1   1 1 1 1
also
hexadezimal   1     7         F
oder dezimal           383
```

Damit beträgt die höchste theoretisch darstellbare Zeilenzahl 384.

Bild 4.5 macht die Verhältnisse nocheinmal grafisch deutlich. Alle Angaben beziehen sich auf niedrigauflösende Grafik ohne Interlace. Bei Benutzung dieser Eigenschaften verdoppeln sich die Werte jeweils!

Daß vom Amiga nur 200 Zeilen genutzt werden, hat seinen Grund in der amerikanischen NTSC-Fernsehnorm, die für ein Vollbild insgesamt 525 Zeilen vorsieht, also für jedes Halbbild etwa 262. Davon gehen 20 Zeilen für die vertikale Austastlücke verloren, weil der Kathodenstrahl in dieser Zeit unsichtbar zum oberen Bildschirmrand zurückläuft. Außerdem werden zusätzlich einige Zeilen auf den verdeckten Rand oberhalb und unterhalb der Bildröhre geschrieben. Im Interlace-Modus verdoppelt sich die Zeilenzahl wieder.

Das hierzulande übliche PAL-System arbeitet zwar mit einer geringfügig niedrigeren Bildwechselfrequenz, stellt dafür aber

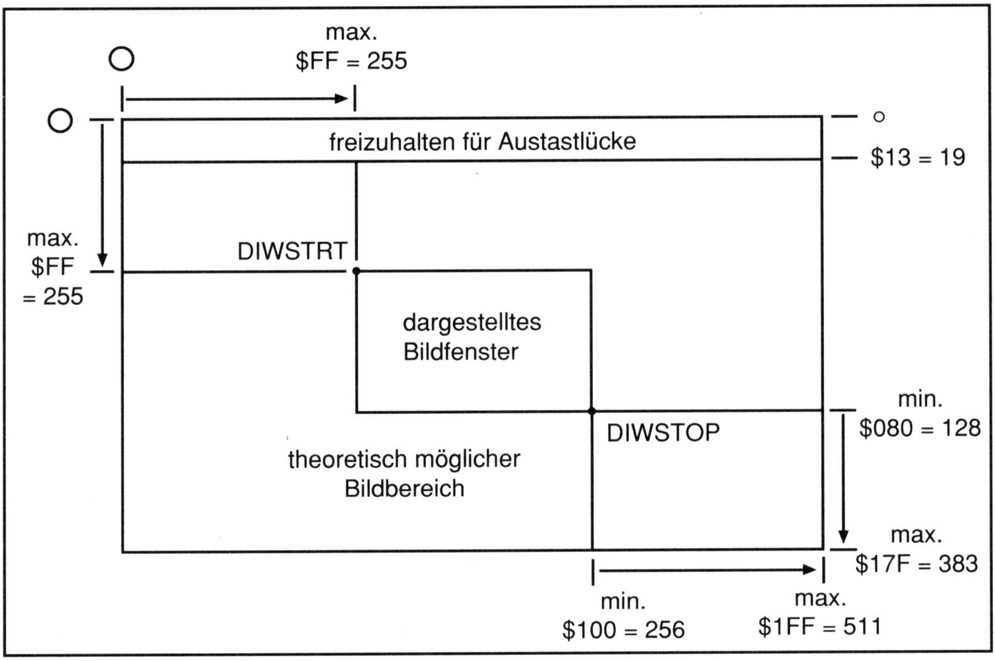


Bild 4.5: Einstellung der dargestellten Bildgröße und Auflösung

mehr Zeilen dar. So enthält ein Vollbild insgesamt 625 Zeilen und damit ein Halbbild etwa 312. Davon gehen wieder 20 Zeilen für den Strahlrücklauf verloren und ebenso einige für die Bildbegrenzung.

Die Standardwerte für die NTSC-Basisbilddarstellung zeigt Tabelle 4.5.

Für europäische Verhältnisse können diese Werte ruhig etwas geändert werden, jedoch ist es zwecklos, die theoretisch maximal mögliche Auflösung einstellen zu wollen. Tabelle 4.5 zeigt, daß auch für die horizontale Ausgabe ein gewisser Vorlauf gewählt wurde. Vor allem für Schriftdarstellung ist es sinnvoll, den Bildschirm-

Richtung	Name	Inhalt	intern	Auflösung
horizontal	HSTART	$ 81	$ 81 = 129	
	HSTOP	$ C1	$1C1 = 449	320 Punkte
vertikal	VSTART	$ 2C	$ 2C = 44	
	VSTOP	$ F4	$0F4 = 244	200 Zeilen

Tabelle 4.5: Standardkoordinaten für ein NTSC-Bild

Randbereich nicht zu benutzen, da hier aufgrund der Mattscheibenkrümmung Unschärfen auftreten können.

Je nachdem, ob hoch- oder niedrigauflösende Grafik angewählt ist, werden die Bildinformationen in jeder Zeile mit einer der beiden festen Geschwindigkeiten ausgegeben. Ist ein Fenster mit mehr Punkten pro Zeile selektiert, dann dauert die Ausgabe entsprechend länger. Weil sich aber der Videostrahl mit fester Geschwindigkeit fortbewegt, wird ein Teil der Informationen wie schon bei der vertikalen Darstellung nicht sichtbar sein. Die optimale Plazierung des Bildes kann sich von Ausgabegerät zu Ausgabegerät unterscheiden. Hier hilft nur experimentieren weiter...

4.2.2 Taktvoll

Die Ausnutzung der vollen Grafikfähigkeiten des Amiga ist nur mit speziellen Monitoren möglich, die ihr Bild nicht nach der Fernsehnorm aufbauen und beispielsweise wesentlich mehr Zeilen darstellen können. Allerdings kostet der Aufbau eines größeren Bildes bei gleicher Pixelgeschwindigkeit mehr Zeit. Es ist wenig sinnvoll, die Bildwechselfrequenz zu senken, denn mit seinen 50Hz befindet sich der Amiga bereits im unteren noch zumutbaren Bereich. Bei weiterer Absenkung ist ein deutliches Flackern des Bildes unvermeidlich.

Also muß die Ausgabegeschwindigkeit des Videoteils erhöht werden. Dann passen ohne weiteres auch mehr Bildpunkte in eine Zeile. Sie wissen bereits, daß die Videoausgabe starr mit der Systemfrequenz gekoppelt ist. Allein wegen der Fernsehnorm beträgt der Prozessortakt nicht volle 8MHz, sondern nur etwa 7,1MHz. Es ist also noch eine gewisse Reserve da, und eine Erhöhung der Taktfrequenz ist sogar von außen möglich!

Im Amiga erzeugt ein Quarzoszillator genau 28,63636MHz. Über Teiler werden daraus die auf der Platine nötigen Frequenzen abgeleitet. Zur Einspeisung von anderen Arbeitsfrequenzen wurden am RGB-Connector die beiden Anschlüsse $\overline{\text{XCLK}}$ und $\overline{\text{XCLKEN}}$ vorgesehen.

Letzterer ist der Auswahleingang (Extern Clock Enable). Bei HIGH wird das intern generierte Taktsignal verarbeitet, bei $\overline{\text{LOW}}$ einfach auf das am Eingang $\overline{\text{XCLK}}$ (Extern Clock) eingespeiste umgeschaltet.

Allein schon die Positionierung der beiden Anschlüsse am RGB-Connector zeigt die vorgesehene Verwendung: Durch eine schlichte Erhöhung der Taktfrequenz läßt sich problemlos ein höher aufgelöstes Bild darstellen. Inzwischen gibt es sogenannte Multisync-Monitore zu erschwinglichen Preisen, die sich selbständig auf die angelegte Zeilen- und Bildfrequenz einstellen, so daß an die externe Quarzfrequenz keine hohen Ansprüche gestellt werden brauchen. Sie sollte allerdings nicht 32MHz überschreiten, da andernfalls eventuell die Spezialbausteine bzw. der Prozessor überfordert wären.

Nebenbei hat diese Schaltung noch den angenehmen Effekt, die Rechenleistung des Amiga zu erhöhen. 32MHz bedeuten eine Steigerung der Arbeitsgeschwindigkeit um 12,7%. Beachten Sie jedoch, daß über die Systemtaktfrequenz auch gewisse

interne Abläufe gesteuert werden, wie etwa die Zeiteinteilung für das Multitasking oder die Stepperrate der Laufwerke. Dabei könnte es eventuell zu Problemen kommen.

Eine andere Verwendungsmöglichkeit der externen Takteinspeisung ist Genlocking. Von einer entsprechenden Oszillatorschaltung könnte der Amiga allein hardwaremäßig auf ein vorhandenes Videosignal synchronisiert werden. Allerdings ist dazu eine recht aufwendige PLL-Schaltung im externen Gerät notwendig, die aus dem Video-Synchronsignal ein phasenstarres Taktsignal der geforderten Frequenz generiert. In der Praxis erweist sich das als recht schwierig. Jedoch besitzt der Amiga dazu bereits andere Vorkehrungen.

4.3 Genlock

Die getrennte Generierung der Synchronsignale durch AGNUS ermöglicht eine leichte Abstimmung der Bildausgabe auf andere Geräte. Hier soll ein Interface vorgestellt werden, das es erlaubt, das Bild einer Video-Signalquelle gleichzeitig mit dem Computerbild auf einem Monitor darzustellen. Die Video-Quelle kann dabei ein Fernsehempfänger, ein Videorecorder, ein Bildplattenspieler, eine Videokamera oder etwas ähnliches sein. Einzige Bedingung: Das Gerät muß einen FBAS- oder CVBS-Ausgang besitzen.

4.3.1 Exaktes Timing

Im Videobereich ist das Mischen mehrerer Signale um vieles aufwendiger als bei Audio-Quellen. Da zu jeder Zeit im Video-Spannungsverlauf ein bestimmter Bildpunkt übermittelt wird, müssen die Signale Zeile für Zeile exakt übereinstimmen.

Am Timing eines Fernsehsenders läßt sich nun einmal so leicht nichts drehen. Also muß der Amiga seine Ausgabegeschwindigkeit an die von der anderen Quelle generierten Synchronsignale starr ankoppeln. Beim Amiga wurde ein solcher Genlock-Betrieb bereits fest vorgesehen.

Hier zeigt sich deutlich, daß er von Grund auf als Multi-Media-Computer konzipiert wurde. So läßt sich durch Setzen von Bit 1 im Bit Plane-Kontrollregister BPLCON0 (Adresse $DFF100) die interne Synchronisation abschalten. $\overline{\text{HSY}}$ und $\overline{\text{VSY}}$ am RGB-Connector werden nun zu Eingängen, über die ein externes Gerät das komplette Timing der Video-Ausgabe übernehmen kann. Dies darf nicht mit der gerade in Kapitel 4.2.2 beschriebenen externen Takteinspeisung verwechselt werden. Lediglich die Zeitpunkte, wann mit der Ausgabe einer Zeile zu beginnen ist, werden in diesem Fall von außen bestimmt. Bei einem Vertikal-Synchronimpuls setzt der Amiga seine Display-Zeiger auf den Bildanfang, und nach jedem Horizontal-Impuls startet er die Ausgabe einer neuen Zeile.

Die Taktfrequenz des Rechners und damit die Zeilenlänge bleibt gleich.

Das Betriebssystem des Amiga prüft nach dem Einschalten, ob Synchronsignale von außen eingespeist werden. Nur wenn dies nicht der Fall ist, aktiviert es die interne Generatoren und schaltet die Synchron-Pins am RGB-Connector auf Ausgang.

4.3.2 Bunt gemischt

Soll das Amiga-Bild regelrecht in den Vordergrund eines anderen Bildes eingeblendet werden, dann ist, während die Informationen vom Amiga kommen, das andere Signal total abzuschalten. Zu diesem Zweck steht am RGB-Connector des Amiga noch das Signal $\overline{ZD}$ (Zero Detect) zur Verfügung. Es kommt vom gleichnamigen DENISE-Anschluß und signalisiert mit LOW, daß augenblicklich die Hintergrundfarbe ausgegeben wird.

Das Mischen der beiden Signalquellen kann auf sehr einfache Weise geschehen, denn dabei kommt uns der Amiga-Monitor (Modell 108X) mit seinen alternativen Eingängen sehr entgegen. Bild 4.6 zeigt die prinzipielle Innenschaltung des Gerätes. Sie werden bereits festgestellt haben, daß sich hinter der Klappe mit den Bedienelementen an der Frontseite des Monitors auch ein Umschalter mit der Beschriftung CVBS/RGB befindet. Drückt man ihn nieder, wird der CVBS-Eingang, also die Cynch-Buchse, an der Geräterückseite aktiviert. CVBS (Composite Video and Burst Signals) ist die englische Bezeichnung für FBAS. Dieser Eingang findet sich im Schaltschema links oben. Die ankommenden Signale werden im Video-Vorverstärker und Farbdecoder aufberei-

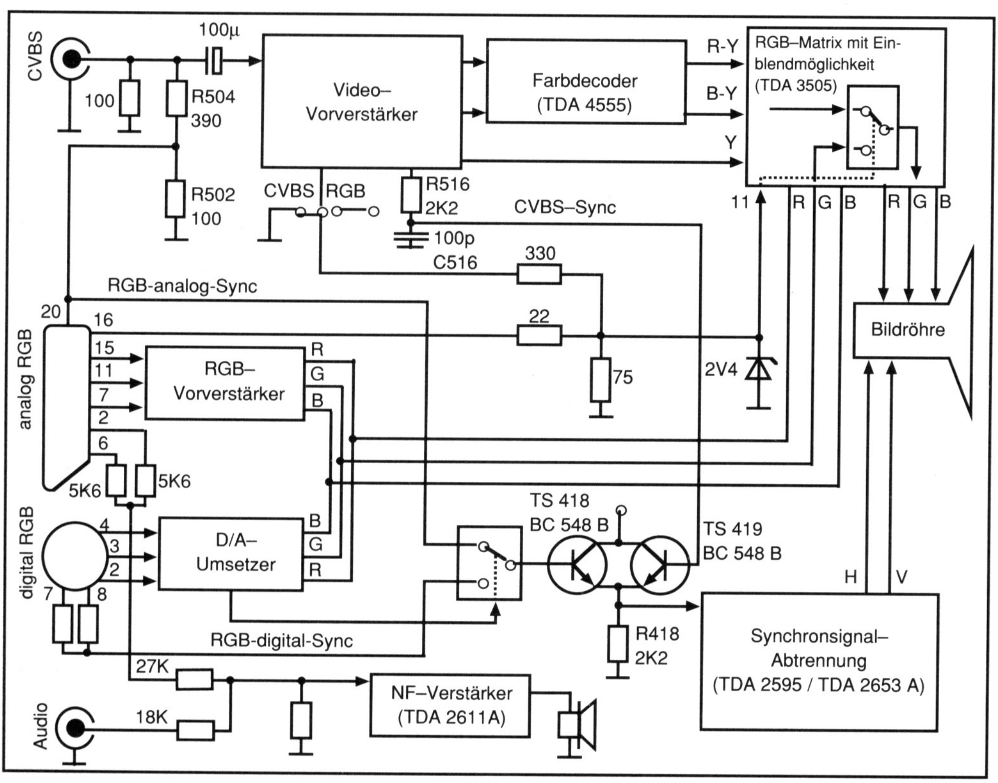


Bild 4.6: Vereinfachte Innenschaltung des Amiga-Monitors 108x

tet und gelangen nach Abtrennung der Syncronsignale zu einer RGB-Matrix.

Dort kommen auch die RGB-Signale vom Computer an. Die Auswahl der darzustellenden Signalquelle geschieht nun mittels eines ICs, das ursprünglich zur Einblendung von RGB-Daten – beispielsweise von einem Videotext-Empfänger – ins Fernsehbild entwickelt wurde. Für Computerbetrieb ist die Normalstellung hier RGB. Dankenswerterweise haben die Entwickler den Umschalt-Anschluß des ICs auch an Pin 16 des SCART-Steckers gelegt. So ergibt sich die Möglichkeit, mittels des Amiga-Hintergrundindikators $\overline{ZD}$ das Fernsehbild schnell auszublenden. Dazu muß $\overline{ZD}$ invertiert werden.

4.3.3 Genlock selbstgebaut

Unter Ausnutzung der im Monitor enthaltenen Mixer läßt sich auf einfache Weise ein preisgünstiges Genlock-Interface aufbauen. Damit ist die Darstellung des gemischten Signals auf dem Monitor möglich, nicht aber die Aufzeichnung auf einen Videorecorder. Zu diesem Zweck müßten die RGB-Signale am Ausgang des TDA 3505 abgegriffen und mit Hilfe eines RGB-FBAS-Wandlers in ein passendes Signal umgesetzt werden. Eine Bauanleitung für diesen Wandler wurde in der Zeitschrift c't 6/87, Seite 68ff veröffentlicht. Allerdigs lassen sich bereits mit dem hier beschriebenen Interface erstaunliche Effekte erzielen.

Ein simples Einsatzbeispiel ist das Zeigen von Details mit einem in das Video-Bild eingeblendeten Pfeil, der über die Amiga-Maus bewegt werden kann.

Bild 4.7 zeigt den Schaltplan des Genlock-Interfaces. Es wird nach Bild 4.8 zwischen den Amiga und seinen Monitor geschaltet und besitzt einen zusätzlichen Cynch-Eingang für das Video-Signal. In der unteren Hälfte des Schaltplans ist die Elektronik zur Abtrennung der Synchronsignale zu erkennen, darüber die Ankopplung an den RGB-Connector des Amiga.

Die Abtrennung der Synchronsignale von einem Fernsehbild ist nicht so einfach, wie man sich das zunächst vorstellen mag, besonders wenn etwa bei Fernsehtunern der Empfang nicht völlig ungestört ist. Grundsätzlich werden die Synchronimpulse über ein Amplitudensieb aus dem FBAS-Signal gefiltert und dann mittels einer Frequenzweiche (Zeitkonstantenglied) voneinander getrennt. Vielerlei Störungen, wie Schwankungen in der Bildhelligkeit, Zündfunken von Motoren oder Haushaltsgeräten und andere Umwelteinflüsse, können dabei allzuleicht Fehlsynchronisierungen verursachen. Besonders die recht niederfrequenten Vertikalsynchronsignale sind davon betroffen. Um eine zuverlässige Impulsabtrennung zu gewährleisten, kommt in dieser Schaltung die integrierte Horizontalkombination TDA 2593 von VALVO zum Einsatz. Sie enthält neben aufwendigen Vorrichtungen zur Störaustastung eine sogenannte Schwungradsynchronisierung. Dabei erzeugt ein Oszillator ständig die Synchronimpulse. Seine Frequenz kann mittels des Reglers V-Hold grob eingestellt werden. Eine Chip-interne PLL-Schaltung stimmt sie dann selbsttätig exakt auf die Synchronsignalfrequenz des eingespeisten Signals ab. So gehen auch bei zeitweisem

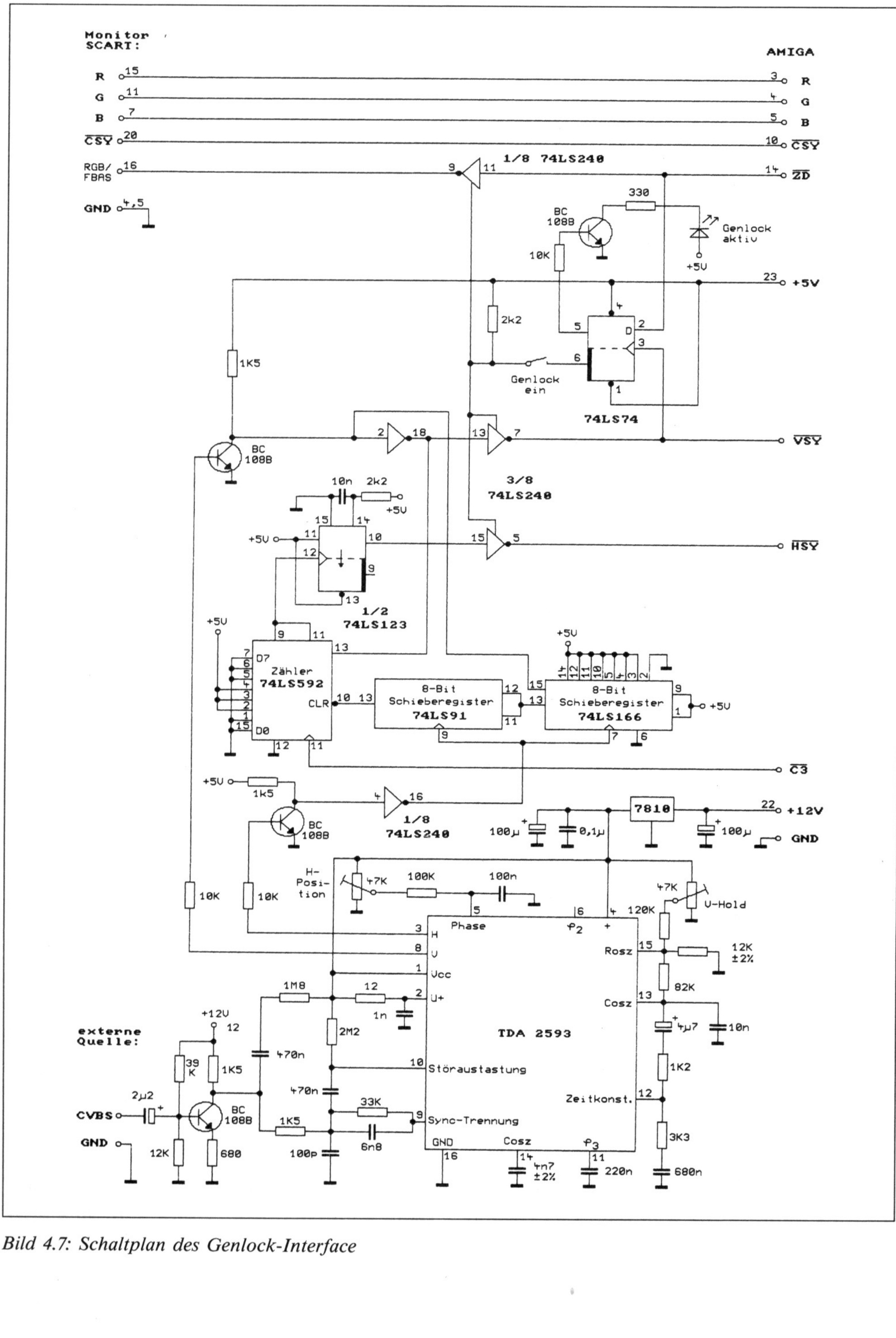


Bild 4.7: Schaltplan des Genlock-Interface

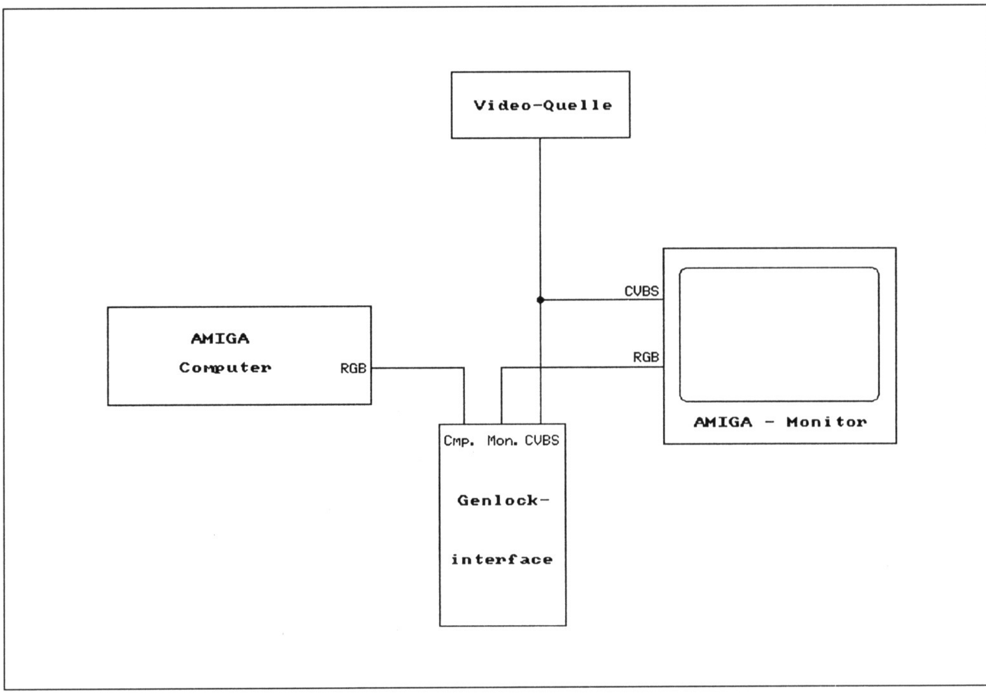


Bild 4.8: Anschluß des Genlock-Interface

Empfangsausfall keine Impulse verloren, was für den Computer besonders wichtig ist. Der TDA 2593 wird hier mit einer stabilisierten Spannung von 10 Volt betrieben, die ein Spannungsregler 7810 aus der 12-Volt-Versorgungsspannung des Amiga gewinnt. Diese stabile Speisespannung ist für den störungsfreien Betrieb unabdingbar.

Da das IC eine positive Lage der Synchronpegel im eingespeisten FBAS-Signal benötigt, wird das Eingangssignal zunächst von einer Transistorstufe invertiert. An Pin 3 und 8 stehen dann Horizontal- sowie Vertikalsynchronsignal ebenfalls als positive Impulse zur Verfügung, allerdings mit einer Amplitude von etwa 10 Volt. Die nachgeschalteten Transistorstufen passen die Ausgangsspannung an TTL-Erfordernisse an. Die dadurch hervorgerufene Invertierung machen die ständig durchgeschalteten, negierenden Treiber aus dem IC 74LS240 wieder wett.

Bei der Beschaltung der SYNC-Anschlüsse am RGB-Connector des Amiga ist sicherzustellen, daß niemals gleichzeitig sowohl die internen als auch die externen Bausteine Ausgänge darstellen. Diese Konstellation könnte einen Schaden an AGNUS hervorrufen. In die entsprechenden Leitungen ist zwar jeweils ein Längswiderstand von 47 Ohm geschaltet, auf seine Schutzwirkung sollte man sich aber besser nicht verlassen. Die Auskopplung

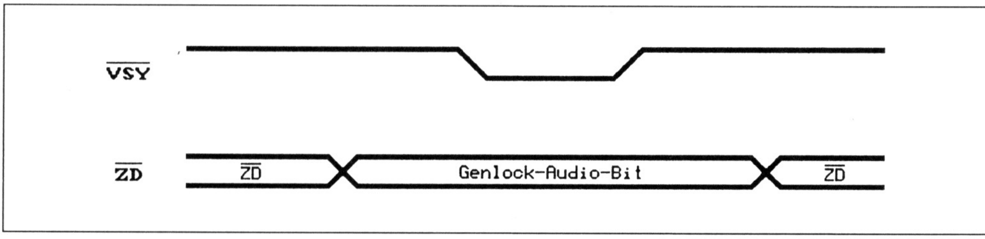


Bild 4.9: Signalisierung des Genlock-Audio-Bits während der Vertikal-Austastlücke

der Synchronsignale erfolgt daher mittels Tristate-Treibern, die über ein Flipflop ein- und ausgeschaltet werden können. Für den Zustand dieses Flipflops ist wie schon für die externe Synchronisierung ein Bit im Bit Plane-Kontrollregister BPLCON0 verantwortlich. Es handelt sich um Bit 8 und ist dort mit GAUD (Genlock Audio Enable) bezeichnet. Sein Zustand wird während der vertikalen Austastlücke nach Bild 4.9 auf den Anschluß $\overline{\text{ZD}}$ gemultiplext. Für eine ordnungsgemäße Funktion dieser Zusatzschaltung müssen immer beide Genlock-Bits gesetzt werden. Wie der Schaltplan zeigt, wird gleichzeitig mit der Umschaltung der Synchronsignalquelle auch das $\overline{\text{ZD}}$-Signal auf den Umschalteingang des Monitors durchgeschaltet.

Damit ein flackerfreies Computerbild entsteht, muß nicht nur die Frequenz mit dem Videobild übereinstimmen, sondern auch dessen Phasenlage. Die Generierung des Video-Rasters im Amiga geschieht digital. Das hat zur Folge, daß nach Eintreffen eines Horizontal-Synchronimpulses vom Genlock-Interface erst beim nächsten Videotakt die Ausgabe einer Zeile gestartet werden kann. Damit entsteht von Zeile zu Zeile ein gewisser zeitlicher Versatz, der sich als Jittern stark bemerkbar macht. Dieser Effekt läßt sich vermeiden, indem die Zeilenanfänge innerhalb eines Halbbildes auf den Computertakt synchronisiert werden. Dazu steht am RGB-Connector das Taktsignal $\overline{\text{C3}}$ mit der 229-fachen Zeilenfrequenz zur Verfügung. Der programmierbare Zähler 74LS592 wird bei jedem Bildanfang mit einem festen Anfangswert geladen.

Danach erzeugt er am Ausgang die starr mit dem Computer-Videotakt gekoppelte Zeilenfrequenz von exakt 15625 Hz, wobei das Monoflop 74LS123 verwertbare Impulse mit einer Länge von etwa 10 Mikrosekunden formt.

Die Synchronisierung des Zählers auf das extern eingespeiste Bildsignal erfolgt erst in der 16. Zeile jedes Halbbildes, damit die PLL im TDA 2593 nach dem Zeilensprung genügend Zeit hat, sich einzuschwingen. Dazu wird mit dem Vertikalsynchronimpuls der Anfangswert 11111110 in das rechte Schieberegister 74LS166 geladen. Jeder Horizontalimpuls vom TDA 2593 schiebt diesen Wert um eine Stelle nach links, wobei eine 1 nachgezogen wird. Nach 16 Takten, also in der 16. Zeile, ist die 0 auch durch das zweite Schieberegister gewandert und bringt den Zähler in die Stellung 0, womit der linken Bildrand festliegt.

1	doppelseitige Platine nach Bild 4.10
1	D-Flipflop 74LS74
1	8-Bit Schieberegister 74LS91
1	2-fach Monoflop 74LS123
1	8-Bit Schieberegister 74LS166
1	2 mal 4-fach Tristate-Treiber 74LS240
1	programmierbarer Zähler 74LS592
1	Sync-Prozessor TDA 2593 (Valvo)
2	IC-Sockel, 14-pol
4	IC-Sockel, 16-pol
1	IC-Sockel, 20-pol
1	Spannungsregler 7810
4	Transistoren BC108B o.ä.
1	Leuchtdiode, Farbe nach Wahl
1	Widerstand 12 Ohm
1	Widerstand 330 Ohm
1	Widerstand 680 Ohm
1	Widerstand 1,2 Kiloohm
4	Widerstände 1,5 Kiloohm
2	Widerstände 2,2 Kiloohm
1	Widerstand 3,3 Kiloohm
3	Widerstände 10 Kiloohm
2	Widerstände 12 Kiloohm
1	Widerstand 39 Kiloohm
1	Widerstand 82 Kiloohm
1	Widerstand 100 Kiloohm
1	Widerstand 120 Kiloohm
1	Widerstand 1,8 Megaohm
1	Widerstand 2,2 Megaohm
2	Potis 47 Kiloohm
1	Kondensator 100 Pikofarad
1	Kondensator 4,7 Nanofarad
1	Kondensator 1 Nanofarad
1	Kondensator 6,8 Nanofarad
2	Kondensatoren 10 Nanofarad
5	Kondensatoren 100 Nanofarad, Keramik
1	Widerstand 33 Kiloohm
1	Kondensator 220 Nanofarad
2	Kondensatoren 470 Nanofarad
1	Kondensator 680 Nanofarad
1	Kondensator 2,2 Mikrofarad/16V
1	Kondensator 4,7 Mikrofarad/16V
1	Kondensator 10 Mikrofarad/16V
2	Kondensatoren 100 Mikrofarad/16V
1	Sub-D-Stecker, 23-polig, weiblich
1	Scart-Stecker
1m	Videokabel
2	Befestigungsschellen
1	Gehäuse Sub-D-Buchse, 23pol
1m	Mini-Koaxialkabel 75 Ohm
1	Chynch-Stecker
1	Chynch-Buchse, lötbar
2	Lötnägel

Tabelle 4.6: Die für das Genlock-Interface benötigten Bauteile

4.3.4 Genlock-Aufbau, Abgleich und Betrieb

Mit der Platine nach Bild 4.10 kann das Genlock-Interface leicht aufgebaut werden. Die Teile aus Tabelle 4.6 werden nach dem Bestückungsplan in Bild 4.11 auf der Platine verlötet. Richten Sie sich nach den Hinweisen in Anhang A. Der TDA 2593 ist ein Standardbauteil für Fernsehgeräte, das in gut sortierten Elektronikläden ohne Schwierigkeiten erhältlich ist. Notfalls können auch die pinkompatiblen Vorgängerversionen TDA 2590 oder TDA 2592 eingesetzt werden. Als Verbindungskabel zum Monitor und zum Computer eignet sich besonders ein Videokabel mit mehreren abgeschirmten 75-Ohm-Leitungen.

Foto 4.1 zeigt das betriebsfertige Gerät. Es wird nach nochmaliger Überprüfung aller Bauteile, Lötstellen und Leiterbahnen nach Bild 4.9 angeschlossen, wobei die

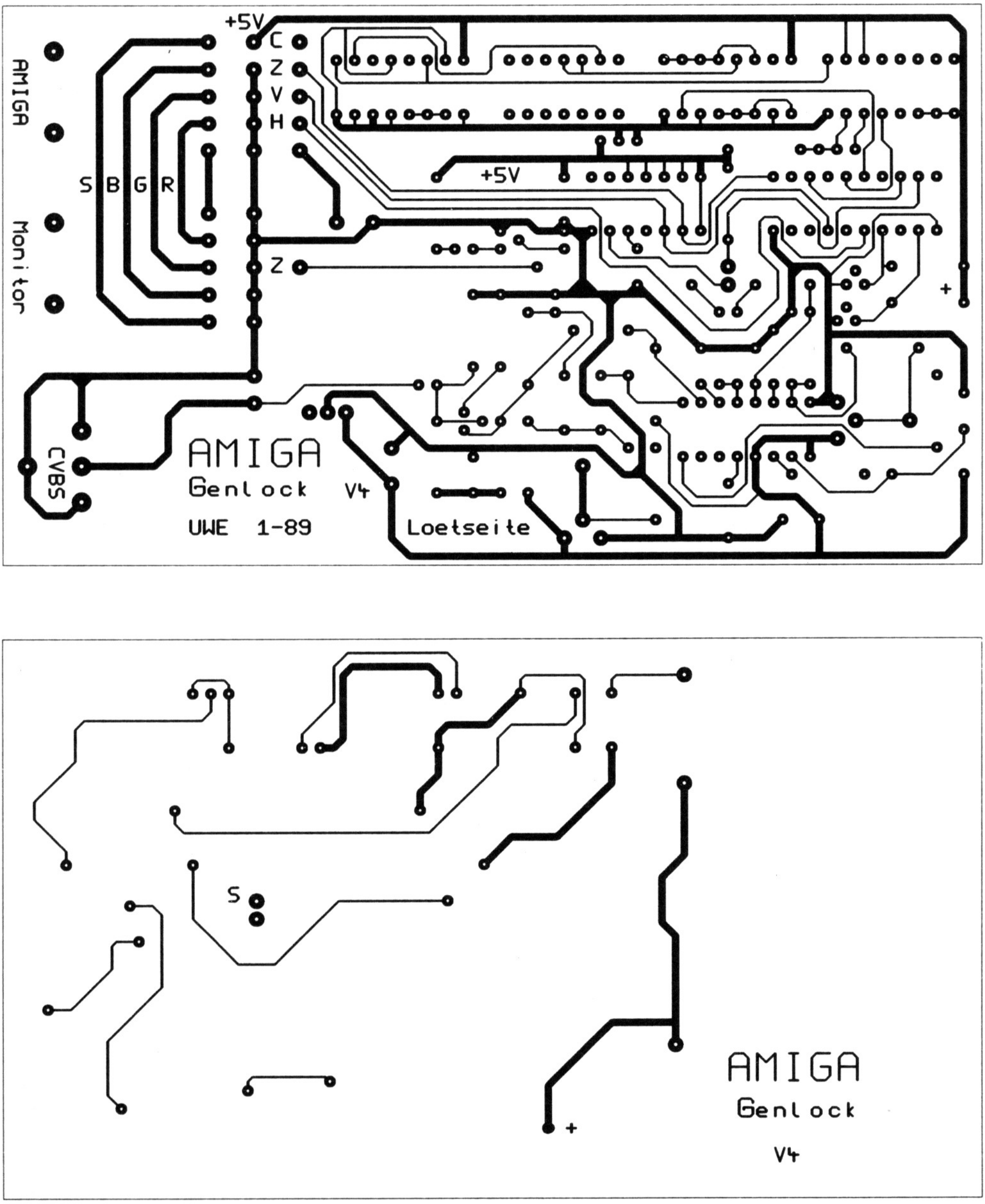


Bild 4.10: Layout für das Genlock-Interface

Video-Quelle mit der Cynch-Buchse zu verbinden ist. Das Kabel zum Monitor kann auch auf der Genlock-Platine am Punkt »Video extern« angelötet werden (Abschirmung an GND).

Für einen ersten Funktionstest wird zunächst nur die Video-Quelle eingeschaltet. Der Wahlschalter im Monitor ist in Stellung CVBS zu bringen. Jetzt muß das Videobild erscheinen. Schließen Sie statt des »Genlock ein«-Schalters an die mit S bezeichnete Klemme (verbunden zum Widerstand) provisorisch ein Kabel nach Masse an. Das Interface wird dadurch immer aktiviert. Schalten Sie nun den Computer ein, legen Sie jedoch vorsichtshalber noch keine Diskette ins Laufwerk. Der Amiga erkennt, daß Synchronsignale von außen anliegen und schaltet in den Genlock-Modus. Das wird durch Aufleuchten der LED auf der Genlock-Platine angezeigt. Zusätzlich zum Videobild sollte auf dem Bildschirm kurz darauf das Computer-Einschaltbild sichtbar werden. Wahrscheinlich sehen Sie nur schnell durchlaufende Störsignale. Verstellen Sie den Regler V-Hold so, daß sich ein stehendes Bild ergibt. Das Amiga-Bild erscheint dabei im Vordergrund; im übrigen Bereich ist das eingespeiste Videobild zu sehen.

Nach diesem Test schalten Sie den Computer wieder aus, entfernen die provisorische Drahtbrücke nach Masse und schließen den Schalter »Genlock-ein« nach dem Bestückungsplan an. Da am Video-Stecker leider kein Reset-Signal zu finden ist, kann das Flipflop bei einem

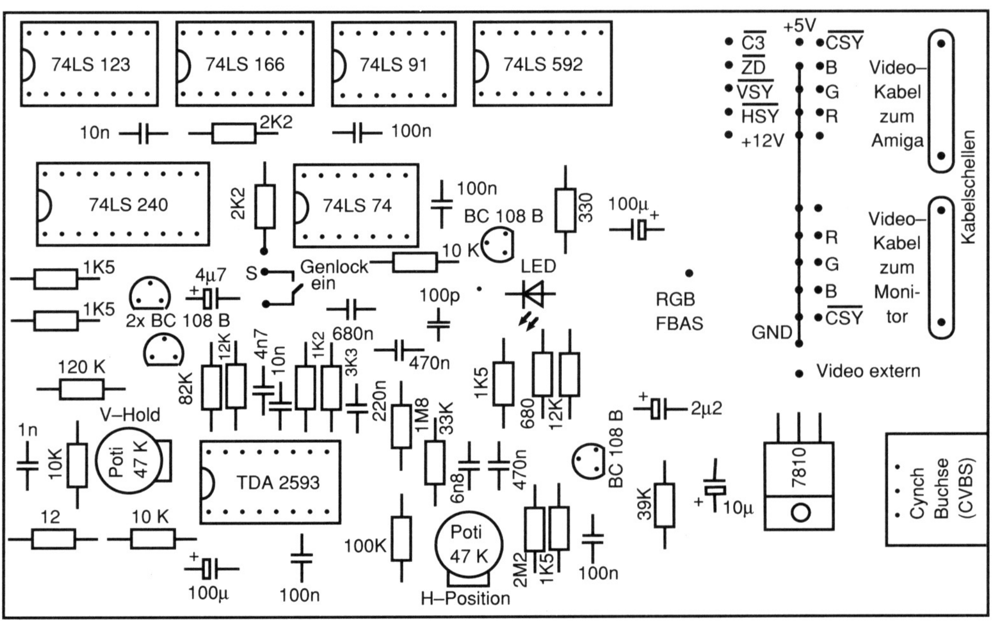


Bild 4.11: Bestückung des Genlock-Interface

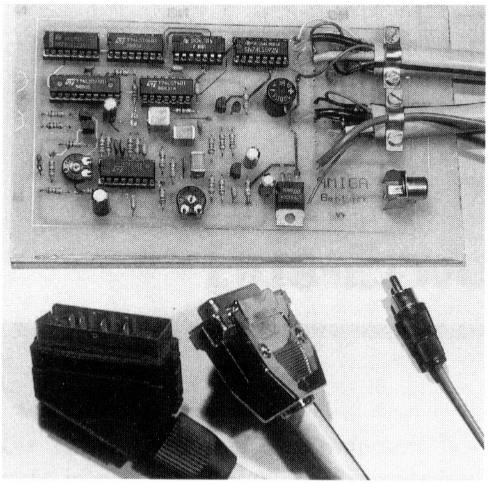


Foto 4.1: Genlock-Interface

Reset nicht in eine feste Ausgangsposition gebracht werden. Es ist also unsicher, ob das Genlock-Interface aktiviert wird oder nicht. Daher wurde der Flipflop-Ausgang über einen Schalter geführt, mit dem sich das Interface ganz abschalten läßt. Dieser Schalter sollte beim Einschalten des Computers oder bei einem Reset durch die Tastenkombination Amiga/Amiga/CTRL immer offen sein. Der Rechner verhält sich dann wie gewohnt.

Probieren Sie das aus, indem Sie den RGB/CVBS-Schalter am Monitor wieder in Stellung RGB bringen, den Genlock-Schalter öffnen, eine Diskette einlegen und den Rechner wie gewohnt hochfahren. Der Schalter auf der Genlock-Platine kann nun geschlossen werden. Zum Betrieb im Genlock-Modus ist das Programm Genlock aus der Diskette zum Buch aufzurufen. Die LED auf der Genlock-Platine leuchtet auf und das Monitor-Bild wechselt in den Interlace-Modus. Drücken Sie nun den RGB/CVBS-Schalter am Monitor. Es erscheint das Videobild im Hintergrund! Das Computerbild kann mit dem Genlock-Regler H-Position in die Mitte des Schirms geschoben werden.

Durch erneuten Aufruf von Genlock erlischt die LED und das eingeblendete Computerbild verschwindet aus dem Videobild. Die Treiber im Genlock-Interface schalten die externen Signale ab, und der Computer aktiviert wieder seine internen Generatoren. Auf diese Weise kann beliebig oft softwaremäßig zwischen den beiden Betriebsarten umgeschaltet werden. Dies ist jedoch nur bei offenem Genlock-Schalter möglich.

5

Bus-Erweiterungen

Die Hauptaufgabe eines Computers besteht in der zielgerichteten Manipulation von Daten. Für einen verwechselungsfreien Zugriff auf große Datenmengen wird in Computersystemen jeder Information eine feste Adresse zugeordnet. Soll eine

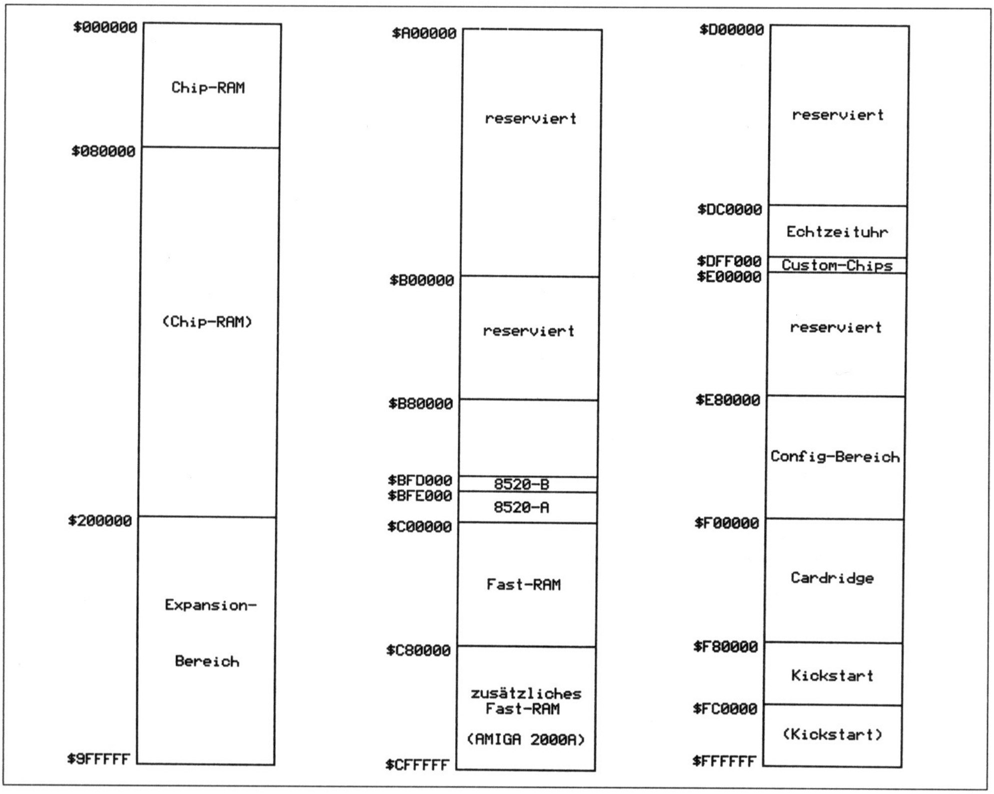


Bild 5.1: Belegung des Amiga-Adreßraums

bestimmte Information verarbeitet, also gelesen oder verändert werden, dann muß die Zentraleinheit dem Speicherinterface deren Adresse sowie die gewünschte Operation mitteilen. Dies geschieht über gemeinsame Adreß-, Daten- und Steuerleitungen, den sogenannten System-Bus. Natürlich wurde auch daraus beim Amiga ein ausgeklügeltes System, das in diesem Kapitel beschrieben werden soll.

5.1 Die Amiga-Infrastruktur

Der Amiga wird mit dem Motorola-Prozessor MC68000 betrieben. Dabei handelt es sich um eine leistungsfähige 16/32-Bit-CPU, bei der neben dem 16 Bit breiten Datenbus auch die Adreßleitungen A1 bis A23 getrennt herausgeführt sind. Da A0 fehlt, werden stets Speicherworte zu je 16 Bit adressiert. Es sind jedoch auch Zugriffe auf 8-Bit-Worte möglich.

Die verschiedenen Bausteine und der interne Speichervorrat der Amiga-Computer belegen längst nicht den gesamten 16 Megabyte großen Adreßraum. Bild 5.1 zeigt die genaue Aufteilung. Einige Bereiche sind für spätere Erweiterungen reserviert und sollten nicht benutzt werden. Fast alle Bausteine sind unvollständig dekodiert. Daher belegen sie einen

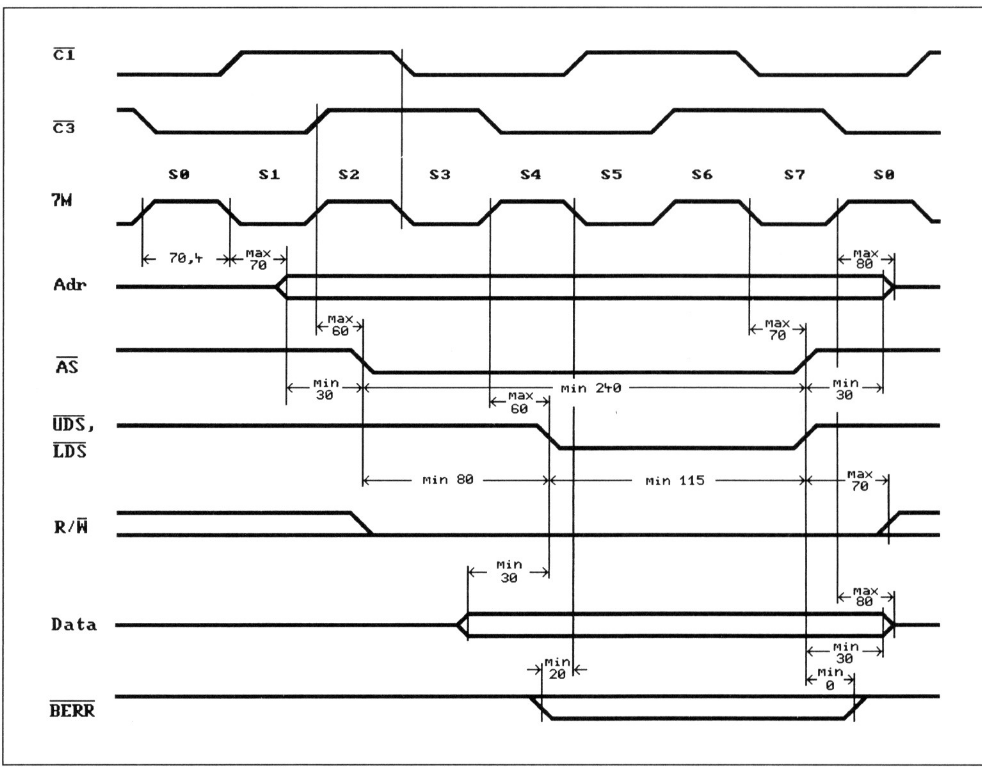


Bild 5.2: 6800 Schreibzyklus

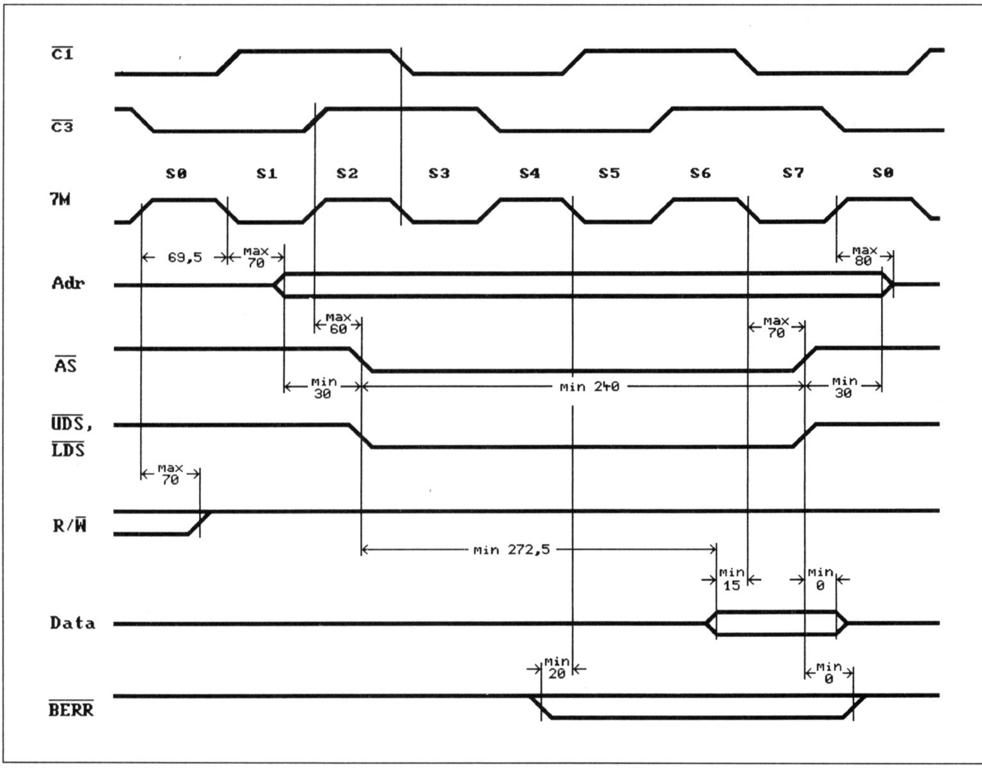


Bild 5.3: 6800 Lesezyklus

wesentlich größeren Adreßbereich als nötig. Ihre Register tauchen außer auf der Basisadresse im gesamten angegebenen Bereich regelmäßig wiederkehrend auf. Dies trifft auch für das Chip-RAM und das Kickstart-ROM zu. Die entstehenden »Spiegelbereiche« sind in Klammern angegeben. Bei neueren Chip-Versionen können diese Bereiche eventuell ebenfalls genutzt werden.

5.1.1 Ablauf von Speicherzugriffen

Die Bilder 5.2 und 5.3 geben einen Überblick zum genauen Zeitverhalten der wichtigsten Prozessorsignale während normaler Speicherzugriffe. Der Prozessortakt 7M (7,1MHz) liefert die Zeitbasis für alle Aktionen. Die Taktphasen sind mit S0 bis S7 bezeichnet. Alle Zeitangaben verstehen sich in Nanosekunden.

Bild 5.2 zeigt die Abläufe, wenn der Prozessor einen Wert ausgibt. Maximal 70ns nach der fallenden Flanke von S0 legt er die Adresse auf den Bus. Danach wird den nachgeschalteten Einheiten mindestens 20ns Zeit gelassen, die Spannungspegel sicher durchzuschalten, bevor der Adreß-Strobe (negative Flanke von $\overline{AS}$) folgt. Gleichzeitig zeigt R/$\overline{W}$ (Read/Write) mit LOW an, daß es sich um einen Schreibzyklus handelt. Minimal 80ns später folgt

der Data-Strobe, indem $\overline{UDS}$ (Upper Data Strobe) nach LOW geht, falls es sich um einen Zugriff auf das höherwertige Datenbyte (D8..D15) handelt, bzw. $\overline{LDS}$ für das niederwertige Datenbyte (D0..D7). Bei einem wortbreiten Zugriff (D0..D15) gehen beide Signale nach LOW. Das Datenwort liegt zu diesem Zeitpunkt bereits mindestens seit 30ns an, so daß es auch taktpegelgesteuerte Bausteine problemlos übernehmen können.

Erst maximal 70ns nach der fallenden Flanke von S6 nimmt der Prozessor die Adreß- und Data-Strobes wieder zurück. Die Adressen und Daten bleiben aber noch für mindestens 30ns stabil. Diese »Hold-Zeit« ist zur sicheren Datenübernahme bei vielen Bausteinen besonders wichtig. Spätestens 80ns nach Beginn des nächsten Zyklus sind die Busse wieder vollständig freigegeben.

Hardware-Fehler, zum Beispiel das gleichzeitige Antworten von mehreren Bausteinen auf einen Speicherzyklus, können eine Ausnahmebehandlung einleiten, wenn mindestens 20ns vor der fallenden Flanke von S4 das Signal $\overline{BERR}$ (Bus Error) gesetzt wird. Beim Amiga hat das normalerweise einen Guru zur Folge.

Bild 5.3 zeigt die Abläufe beim Lesen eines Speicherwortes in den Prozessor. Hier zeigt R/-W bereits 70ns nach dem Beginn des Zyklus mit HIGH an, daß es sich um einen Lesezugriff handelt. Data-Strobe ($\overline{UDS}$ / $\overline{LDS}$) wird nun gleichzeitig mit dem Adreß-Strobe $\overline{AS}$ angelegt. Die Datenübernahme erfolgt während der Taktphase S7. Dazu müssen die Daten mindestens 15ns vor S7 stabil anliegen und dürfen sich erst wieder ändern, nachdem die Strobes zurückgenommen wurden.

5.1.2 Das Betriebssystem-ROM

Am einfachsten kann das Speicherverhalten anhand eines Zugriffs auf das Betriebssystem-ROM erläutert werden. In diesem Baustein ist das zentrale Programm des Amiga gespeichert, das unmittelbar nach dem Einschalten gestartet wird und Ihren Rechner soweit vorbereitet, daß Sie über Maus oder Tastatur Befehle eingeben können. Bei einem ROM handelt es sich um einen Speichertyp, dessen Inhalt nur lesbar ist, im Gegensatz zum RAM aber auch nach dem Abschalten der Betriebsspannung seinen Inhalt nicht verliert.

Bild 5.4 zeigt die Pinbelegung des im Amiga eingesetzten Betriebssystem-ROMs. Es handelt sich um eine Ausführung mit 16 Datenleitungen, 18 Adreßeingängen und zwei Steuerleitungen. Vom Betriebssystem-ROM des Amiga werden nur 17 Adreßleitungen ausgewertet. Es lassen sich daher 2^{17} = 131072 oder 128 Kbyte Speicherworte dressieren (1 Kbyte = 1024 Byte). Jede Adresse enthält entsprechend der Wortbreite des Speicherbausteins 16 Bit. Insgesamt beträgt die Größe des ROMs also 256 Kilobyte oder 2097152 einzelne Speicherzellen.

Daneben sind noch Anschlüsse für die Versorgungsspannung vorhanden. Die Daten- und Adreßleitungen sind direkt mit den entsprechenden Anschlüssen des Prozessors verbunden; die zwei Steuerleitungen $\overline{CS}$ und $\overline{OE}$ werden über eine Aus-

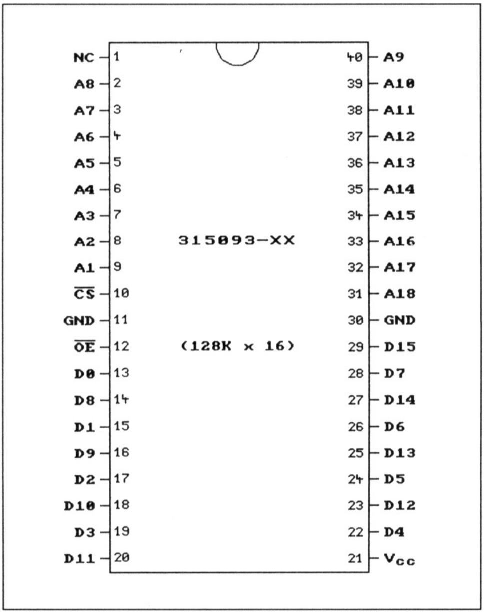


Bild 5.4: Pinbelegung des Kickstart-ROMs (XX steht für Versionsnummer)

wahllogik gesteuert, die dafür sorgt, daß die ROM-Informationen nur auf den Datenbus gelegt werden, wenn eine Adresse im Kickstart-Bereich ($F80000 bis $FC0000, siehe Bild 5.1) anliegt. Im Normalbetrieb sind sie beide inaktiv, also HIGH.

Dem ROM ist zunächst noch alles egal, was um ihn herum passiert, solange sein Steueranschluß $\overline{CE}$ auf HIGH liegt. $\overline{CE}$ ist die Abkürzung für »Chip Enable« (Baustein-Freigabe) und wird manchmal auch mit $\overline{CS}$ für »Chip Select« (Baustein-Auswahl) bezeichnet. Dieser Pin wirkt fast wie ein Ausschalter. Bei HIGH-Pegel befindet sich der Baustein quasi im Winterschlaf. Sogar der Stromverbrauch liegt in dieser »Standby«, »Deselected« oder »Power Down« genannten Betriebsart wesentlich niedriger als im eingeschalteten Zustand. Führt $\overline{CE}$ jedoch LOW-Potential, dann tut sich wenigstens im Inneren des Bausteins etwas: Es wird nachgesehen, welcher Wert unter der mit der augenblicklichen Pegelkombination auf den Adreßleitungen ausgewählten internen Position abgelegt ist; das heißt, eigentlich sind es ja mehrere Werte, die 16 Bits für die 16 Datenleitungen nämlich.

$\overline{OE}$ steht für »Output Enable« (Ausgangs-Freigabe). Wird dieser Anschluß LOW, dann schalten die Daten-Ausgangstreiber ihre Informationen auf die Pins. Um die adressierten Daten an den Ausgängen des ROMs abgreifen zu können, müssen also beide Steueranschlüsse gleichzeitig LOW-Pegel besitzen. Jeder Speicherbaustein benötigt eine gewisse Verzögerungszeit, die vom Anlegen der Adresse bis zum Erscheinen des korrekten Inhalts auf den Datenleitungen vergeht. Es handelt sich um die Zugriffszeit (Access Time), die für die mögliche Arbeitsgeschwindigkeit eines Rechners große Bedeutung hat.

Da sich die im Betriebssystem-ROM befindlichen Informationen nicht ändern lassen, muß dieser Baustein bei Verwendung neuer Software ausgetauscht werden. Lange Zeit wurden die Amiga-Computer mit dem Betriebssystem Kickstart 1.2 ausgerüstet. Seit Oktober 1988 baut Commodore in die neu hergestellten Rechner jedoch die verbesserte und in mehreren Punkten sogar vollständig geänderte Version Kickstart 1.3 ein. Mit welcher Version Ihr Computer arbeitet, läßt sich leicht anhand der Einschaltmeldung ermitteln. Um in den Genuß der neuen Version zu kommen, muß das alte ROM

gegen ein neues ausgetauscht werden, das über jeden guten Fachhändler zu beziehen ist.

Es ist absehbar, daß nicht jede für Kickstart 1.2 geschriebene Software auch unter dem neuen Kickstart 1.3 problemlos läuft, so daß in solchen Fällen ein Wechsel der ROMs nötig wird. Das damit verbundene Öffnen des Geräts ist umständlich und nicht jedermanns Sache. Abgesehen davon werden die ROMs durch das vielfache Aushebeln und Einstecken auch nicht besser. Es muß also ein Adapter her, der es ermöglicht, beide ROMs umschaltbar im Rechner anzuschließen. Diese Forderung erfüllt die im folgenden beschriebene Umschaltplatine Kicki. Anschließend wird noch eine Umschaltplatine vorgestellt, bei der das alternative Betriebssystem in EPROMs – also auch nach eigenen Vorstellungen modifiziert – untergebracht werden kann.

5.1.2.1 Kickstart-ROM-Umschaltung mit Kicki

Foto 5.1 zeigt die kleine Kickstart-ROM-Umschaltplatine Kicki. Zum Betrieb wird der Original-ROM-Baustein aus seinem Sockel entfernt und stattdessen die vorgestellte Platine dort eingesteckt. Bild 5.5 zeigt den entsprechenden Schaltplan. Alle nötigen Signale werden dem Original-Sockel entnommen und auf der Platine zum Anschluß der beiden Zusatzsockel aufbereitet.

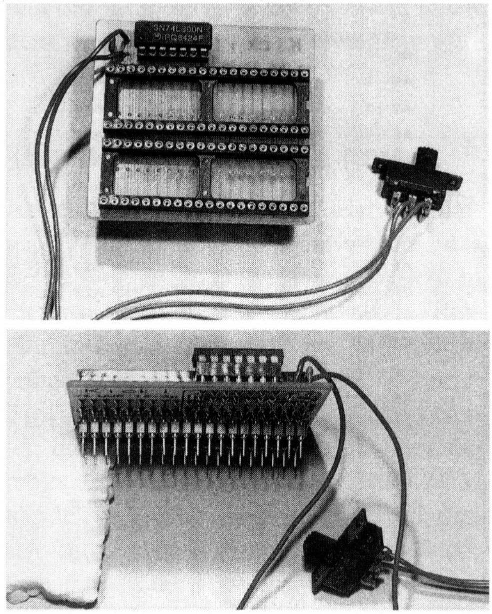


Foto 5.1: »Kicki«

Von der Konstruktion her muß vor allem ausgeschlossen sein, daß beide Bausteine gleichzeitig Informationen auf eine Datenleitung legen, denn falls die Ausgangsspannungen unterschiedlich sein sollten, ähnelt das einem Tauziehen, und der schwächere der beiden Ausgangstreiber läuft Gefahr, wegen Überlastung zerstört zu werden. Die Auswahl geschieht mittels der ROM-Steuerleitung $\overline{CS}$. Vom Original-Sockel auf der Amiga-Mutterplatine wird das Original $\overline{CS}$-Signal dem NAND-Baustein 74LS00 zugeführt und zunächst invertiert. Dann gelangt es an zwei weitere Gatter, deren Ausgang jeweils den $\overline{CS}$-Eingang eines der ROMs treibt. Der zweite Eingang dieser Gatter ist durch die Schalterstellung festgelegt. Bei offenem Schalter entsteht durch den Pull-up-Widerstand HIGH-Pegel am Schalter. Damit wird das LOW-aktive Select-Signal zum linken Kicki-ROM-Sockel durchgeschaltet, während der Eingang des zweiten Gatters LOW erhält und sein Ausgang damit durchgehend auf HIGH bleibt. Bei

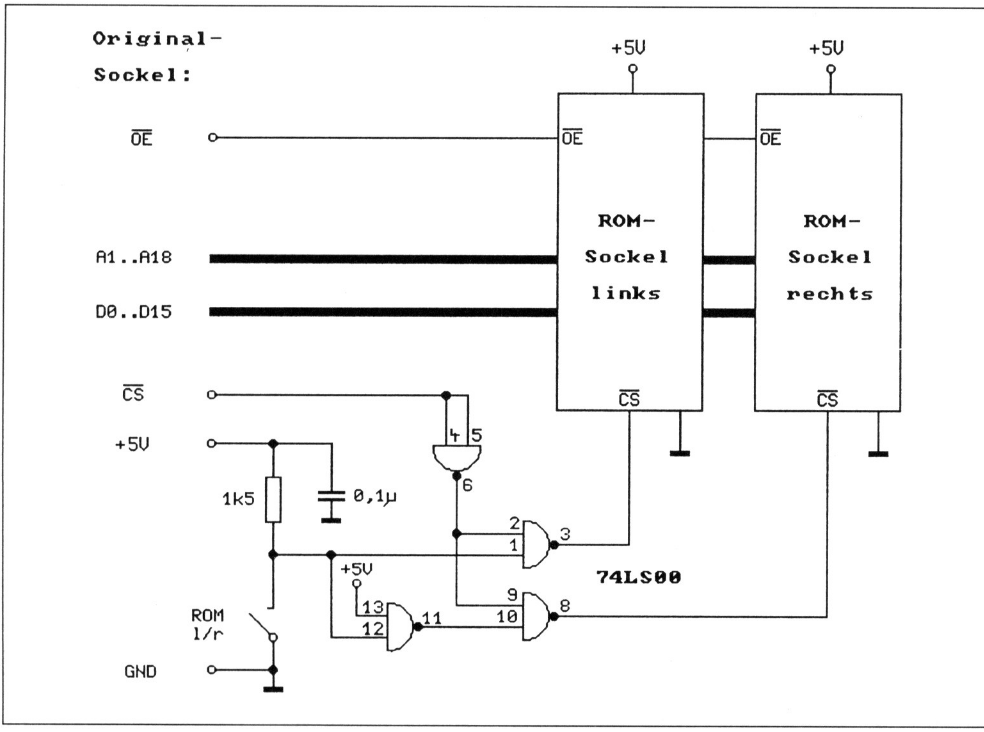


Bild 5.5: Die Amiga-ROM-Umschaltplatine Kicki

geschlossenem Schalter kehren sich die Verhältnisse um.

Natürlich hätte man auch einfach die Select-Leitungen über einen Umschalter führen können. Lange Leitungen können aber bei den hier auftretenden kurzen und sehr schnell aufeinanderfolgenden Impulsen leicht Störungen hervorrufen. Um den Einbauort frei wählbar zu machen und eine simple zweiadrige Leitung verwenden zu können, wurde die hier vorgestellte Lösung eingesetzt.

Die Einzelteile – zusammengestellt in Tabelle 5.1 – werden wie im Bestückungsplan (Bild 5.6) auf der Platine nach dem Layout in Bild 5.7 verlötet. Dabei ist dar-

1	einseitige Platine nach Bild 5.7
1	Kickstart-ROM V1.3 (Commodore-Ersatzteil Art.Nr. 315093-02)
1	4-fach NAND 74LS00
1	IC-Sockel 14-polig
2	IC-Sockel 40-polig
1	Kontaktleiste mit beidseitigen Pins, mind. 40-polig (z.B. Bürklin DV-Nr. B102.114)
1	Schalter, 1 x ein
	Schaltlitze zur Verbindung mit der Platine
1	Widerstand 1,5 Kiloohm
1	Kondensator 100 Nanofarad (Keramik)

Tabelle 5.1: Der Einkaufszettel für Kicki

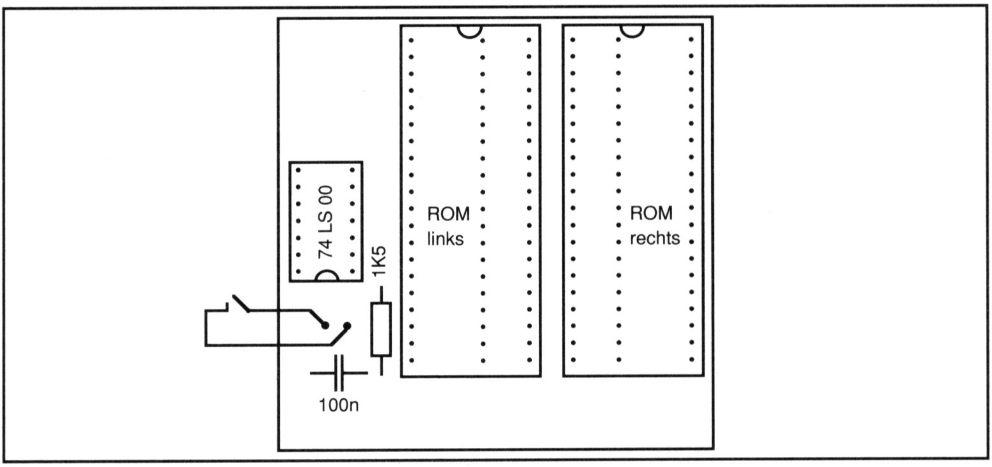


Bild 5.6: Bestückung von Kicki

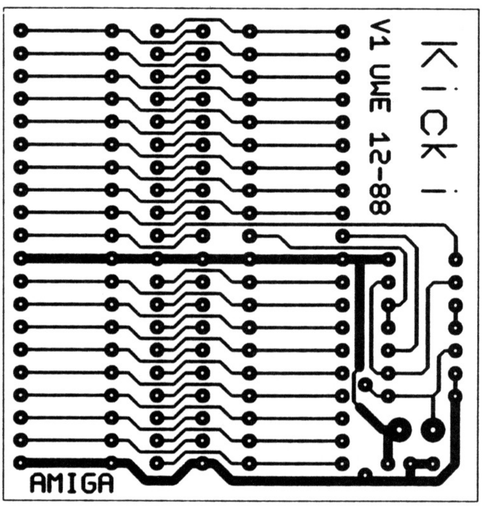


Bild 5.7: Layout von Kicki

auf zu achten, daß die gestrichelt dargestellten Kontaktleisten von der Platinenunterseite her einzusetzen sind. Achtung: Der NAND-Baustein 74LS00 wird mit anderer Orientierung eingesetzt als die beiden ROMs! Bitte den Bestückungsplan genau beachten.

Die Platine paßt sowohl in einen Amiga 500 als auch in einen Amiga 2000. Beim Amiga 500 befindet sich das ROM links unterhalb der Tastatur, beim Amiga 2000 im vorderen Teil unterhalb der Laufwerke, jeweils neben dem Prozessor 68000. Es handelt sich um ein 40-poliges IC und sollte gesockelt sein.

Der Einbau der Umschaltplatine darf nur bei ausgeschaltetem Gerät vorgenommen werden. Bitte richten Sie sich nach den Hinweisen im Anhang A. Beachten Sie auch, daß es sich um MOS-Bauteile handelt. Es wäre sehr ärgerlich, wenn sie Schaden nehmen würden. Kennzeichnen Sie sich deutlich, auf welcher Seite der Mutterplatine die Kerbe für Pin 1 des ROMs liegt. Bei falschem Einsetzen kann ein Bauteil schnell zerstört werden.

Hebeln Sie nun den Original-ROM-Baustein mit einem Schraubenzieher vorsichtig aus seinem Sockel, und stecken Sie ihn sofort in die 28-polige Fassung auf der bereits fertig bestückten Zusatzplatine.

Die Kerbe im Gehäuse muß sich dabei dort befinden, wo sie auch im Bestückungsplan (Bild 5.6) markiert ist. Anschließend wird die Zusatzplatine in die jetzt freie Originalfassung eingesteckt, natürlich so, daß die Kerbe im Gehäuse des ROMs wieder in dieselbe Richtung zeigt wie vorher.

Schalterstellung	Funktion
ein	rechtes ROM aktiviert
aus	linkes ROM aktiviert

Tabelle 5.2: Die Wirkung des Umschalters

Nach dem Einschalten muß das System dann reagieren wie gewohnt. Mit dem Schalter läßt sich zwischen den beiden ROMs umschalten. Tabelle 5.2 zeigt seine Funktion.

Die Länge der Verbindungskabel zum Schalter ist grundsätzlich frei wählbar. Statt mit einem Schalter kann die Auswahl des aktiven ROMs auch über eine logische Spannung beispielsweise von einer anderen Zusatzschaltung her erfolgen.

5.1.2.2 Kickstart in EPROMs

Natürlich läßt sich das neue Betriebssystem auch in EPROMs brennen. Dieses Vorgehen ist dann sinnvoll, wenn gleich einige Änderungen am System vorgenommen werden sollen, wie etwa volle Größe des ersten Bildschirmfensters, optimale Einstellung der Disk-Stepzeit, modifizierte Requester und Fehlertexte usw.

Das Betriebssystem benötigt einen Speicherbereich von insgesamt 256 Kilobyte, der für das Vorhaben in EPROMs untergebracht werden muß. Zur Zeit ist das günstigste verfügbare EPROM dieser Größenordnung vom Typ 27512. Es stellt 64 Kilobyte bei einer Wortbreite von acht Bit zur Verfügung. Zur Unterbringung des Betriebssystems werden damit vier Stück benötigt.

Bild 5.8 zeigt den Schaltplan der EPROM-Platine. Ähnlich wie bei der zuvor beschriebenen Umschaltplatine Kicki kommen auch hier alle Signale vom Original-ROM-Sockel. Allerdings existieren anstelle des zweiten ROMs hier vier EPROMs. Das Steuersignal $\overline{OE}$ wurde allen Einheiten direkt zugeführt. Jedoch schaltet nur diejenige Einheit ihre Datenausgänge durch, deren Steuersignal $\overline{CS}$ gleichzeitig LOW-Pegel führt. Dieses Signal wird daher in Abhängigkeit von der Schalterstellung für die Bausteine getrennt erzeugt.

Ist der Schalter geschlossen, liegt an Pin 10 des Oder-Bausteins 74LS32 LOW, und das $\overline{CS}$-Signal vom Original-Sockel wird an das auf der Platine eingesteckte ROM weitergeleitet, während der Ausgang des anderen Oder-Gatters konstant HIGH-Pegel führt. Damit sind auch alle vier EPROMs durch HIGH an $\overline{CS}$ inaktiv.

Im umgekehrten Fall bleibt das ROM gesperrt, während die Adreßleitung A17 bestimmt, welche EPROMs aktiviert werden. Da es sich um 8-Bit-EPROMs handelt, bilden immer die Inhalte aus zwei Bausteinen ein vollständiges 16-Bit-Wort. Der Zustand von A17 entscheidet, ob die EPROMs mit der oberen oder unteren Betriebssystemhälfte angesprochen werden.

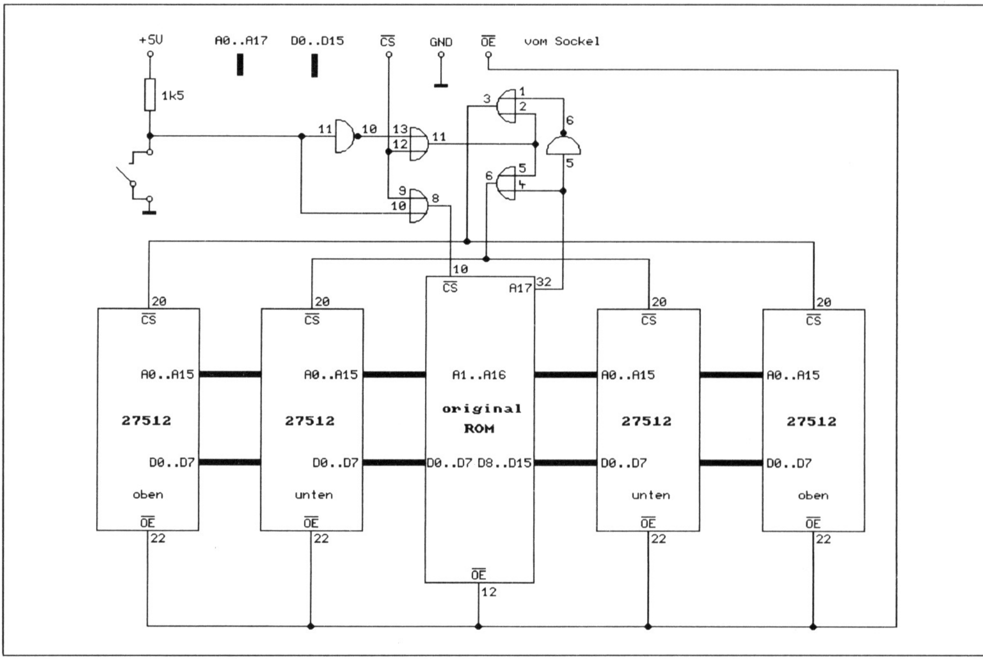


Bild 5.8: Kickstart-Umschaltung mit EPROMs

Da die Unterscheidung auf gerade (even) oder ungerade (odd) Adresse beim 16-Bit-Prozessor 68000 mittels zweier Strobe-Leitungen erfolgt, gibt es dort keine Adreßleitung A0.

Daher wurden die Adreßleitungen A0 der EPROMs über den ROM-Sockel an A1 des Prozessors geschaltet, A1 an A2 und so fort.

Zur Unterbringung der Betriebssystem-Daten in den vier EPROMs muß deren Verteilung genau beachtet werden. Wegen der Aufteilung von High- und Lowbyte ist es einerseits nötig, nach geraden und ungeraden Adressen aufzuteilen. Andererseits muß nach oberer und unterer Adreßraumhälfte unterschieden werden. Tabelle 5.3 zeigt die genaue Zuordnung.

Inhalt EPROM	D8..D15 (even-Byte)	D0..D7 (odd-Byte)
»unten«	$F80000 $F80002 . . . $F9FFFC $F9FFFE	$F80001 $F80003 . . . $F9FFFD $F9FFFF
»oben«	$FA0000 $FA0002 . . . $FBFFFC $FBFFFE	$FA0001 $FA0003 . . . $FBFFFD $FBFFFF

Tabelle 5.3: Zuordnung der Kickstart-Daten zu den EPROMs

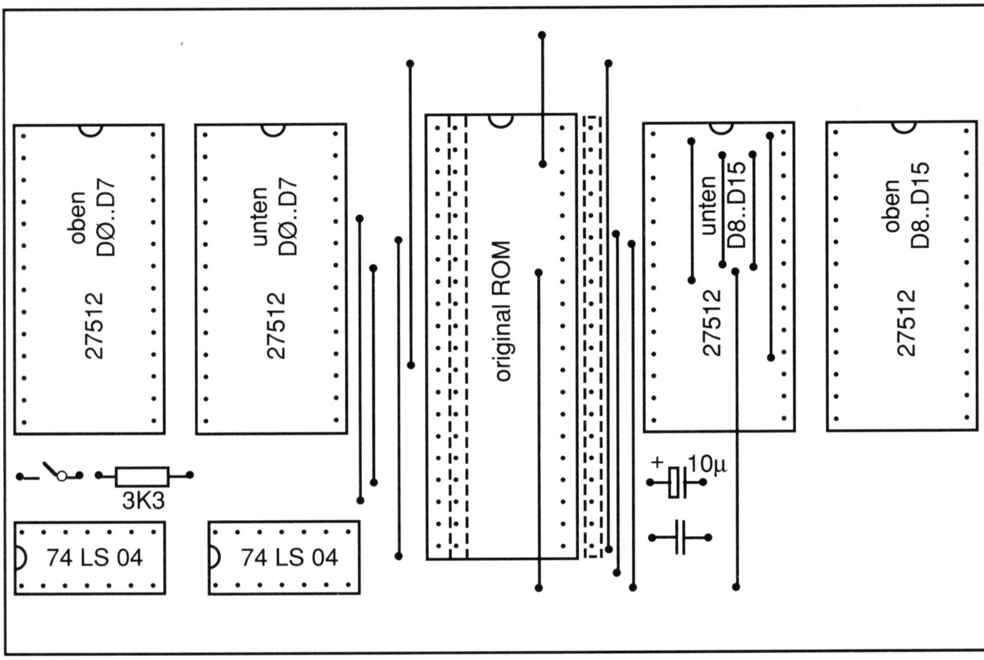


Bild 5.9: Bestückung der EPROM-Platine

Für die EPROMs sind Ausführungen mit einer Zugriffszeit von 200 Nanosekunden oder weniger zu verwenden.

Die zum Aufbau der Umschaltplatine benötigten Einzelteile wurden in Tabelle 5.4 zusammengestellt. Sie werden nach dem Bestückungsplan (Bild 5.9) auf der Platine nach dem Layout in Bild 5.10 verlötet. Richten Sie sich nach den allgemeinen Hinweisen im Anhang A. Zunächst sind die 16 Lötbrücken einzusetzen, die teilweise unter den IC-Sockeln verlaufen und später nur sehr schwer zugänglich sind. Besonders ist auf den richtigen Einbau der gestrichelt dargestellten Kontaktleisten von der Platinenunterseite her zu achten. Foto 5.2 zeigt die betriebsfertige Platine.

1	einseitige Platine nach Bild 5.10
4	EPROMs 27512 (max 200ns)
1	6-fach NOT 74LS04
1	4-fach OR 74LS32
2	IC-Sockel 14-polig
4	IC-Sockel 28-polig
1	IC-Sockel 40-polig
1	Kontaktleiste mit beidseitigen Pins, mind. 40-polig (z.B. Bürklin DV-Nr. B102.114)
1	Schalter, 1 x ein
	Schaltlitze zur Verbindung mit der Platine
1	Widerstand 3,3 Kiloohm
1	Kondensator 100 Nanofarad (Keramik)
1	Elektrolytkondensator 10 Mikrofarad / 16 Volt
2	Lötnägel

Tabelle 5.4: Die Einzelteile für die EPROM-Umschaltplatine

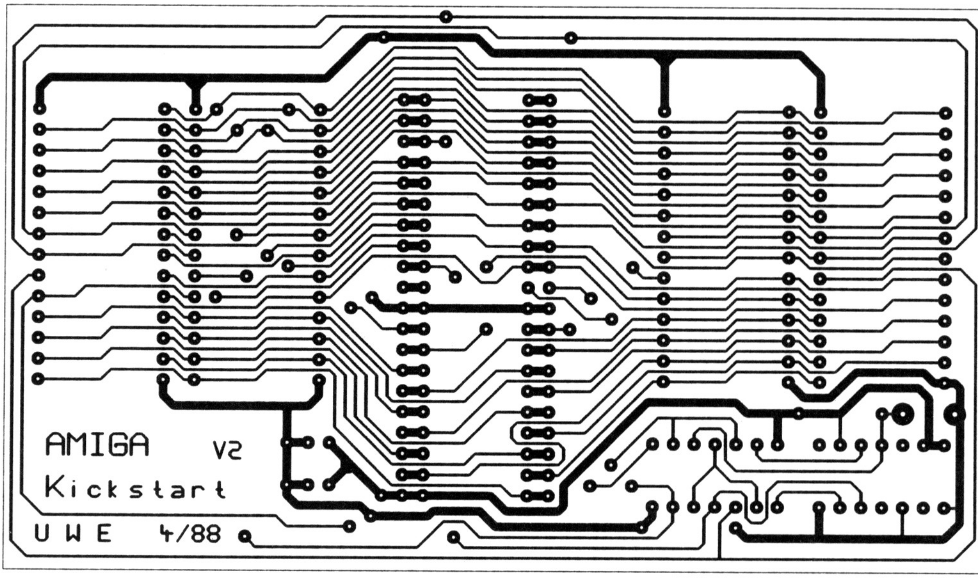
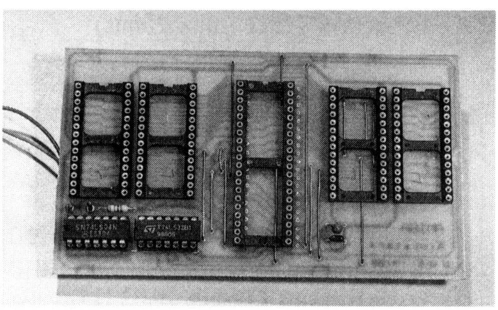


Bild 5.10: Layout der Umschaltplatine für Kickstart in EPROMs



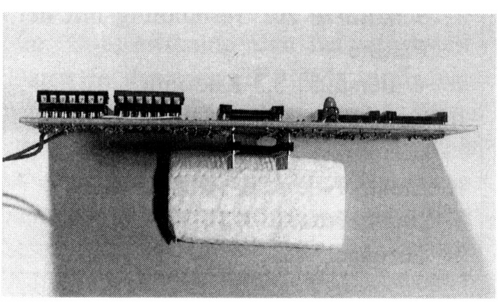


Foto 5.2: EPROM-Umschaltplatine

Auch diese Umschaltplatine paßt sowohl in einen Amiga 500 als auch in einen Amiga 2000. Der Einbau erfolgt genau wie bei der ROM-Umschaltplatine Kicki beschrieben. Tabelle 5.5 zeigt die hier wirksame Funktion des Umschalters. Ebenso läßt sich die Umschaltung mit einer logischen Spannung verwirklichen.

Schalterstellung	Funktion
ein	ROM aktiviert
aus	EPROMs aktiviert

Tabelle 5.5: Die Wirkung des Umschalters

5.2 Der Expansion-Bus

Die Konstruktion der ROM-Umschaltplatinen hat bereits gezeigt, wie wirkungsvoll sich Systemeigenschaften beim Zu-

griff auf Prozessorsignale manipulieren lassen. Über Bus-Erweiterungs-Steckverbindungen hat man bei allen Amiga-Computern Zugriff auf die wichtigen Systemsignale. Die Rechner sind damit offen für jede denkbare Erweiterung. Optimal ist natürlich die Ausführung beim Amiga 2000 mit seinem Slotsystem, aber auch die »kleinen« lassen sich mit Zusatzkarten aufrüsten. Dabei besteht grundsätzlich kein Unterschied zwischen den einzelnen Erweiterungssteckverbindungen. Der Teufel steckt allerdings wie immer im Detail.

5.2.1 Amiga-Infrastruktur

Im Amiga 2000 sind eine Vielzahl von Slots eingebaut, die eine einfache Erweiterbarkeit durch simples Einstecken von Zusatzkarten erlauben. Bild 5.11 zeigt die Anordnung der einzelnen Slots. Wenn der geöffnete Amiga 2000 mit der Frontseite zum Betrachter steht, befinden sich links vorne die fünf Amiga-Slots, gleich dahinter die kurzen AT-Slots und daran anschließend die vier PC-Slots.

Die Bauteilseite ist nach rechts zu den Diskettenlaufwerken hin gerichtet. Pin 1

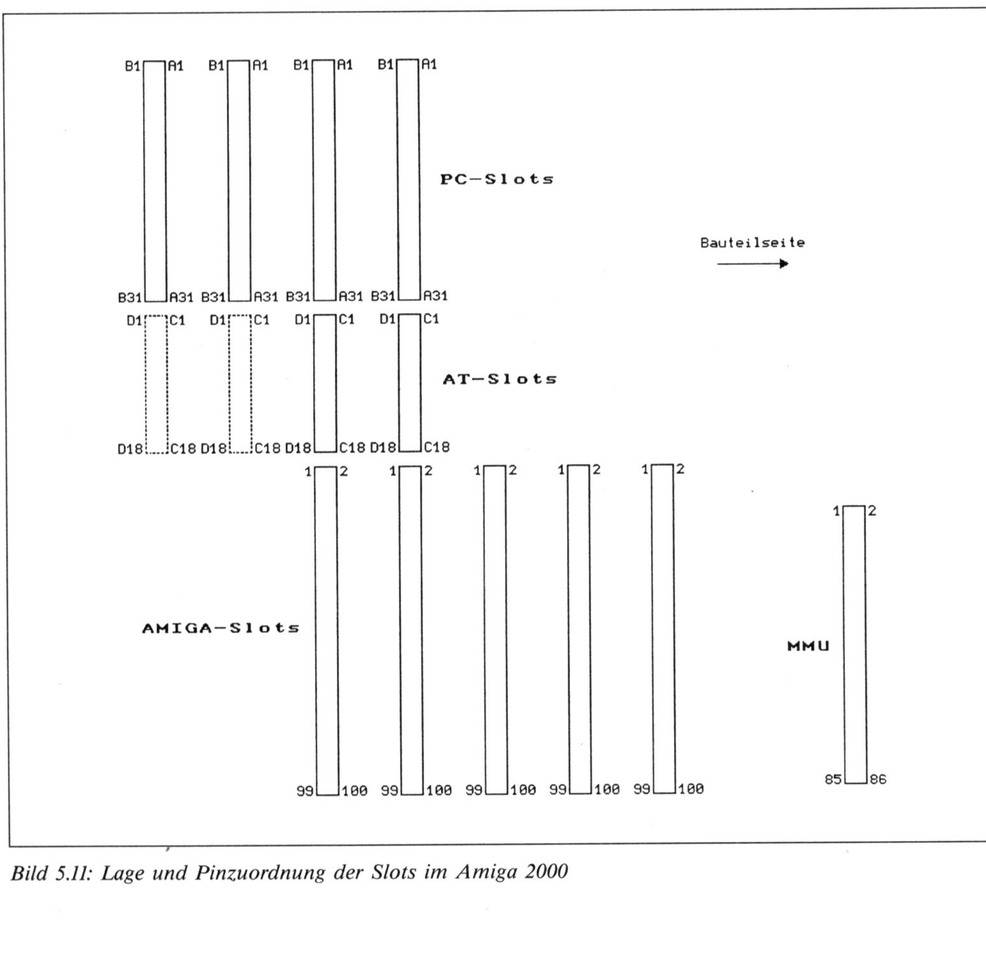


Bild 5.11: Lage und Pinzuordnung der Slots im Amiga 2000

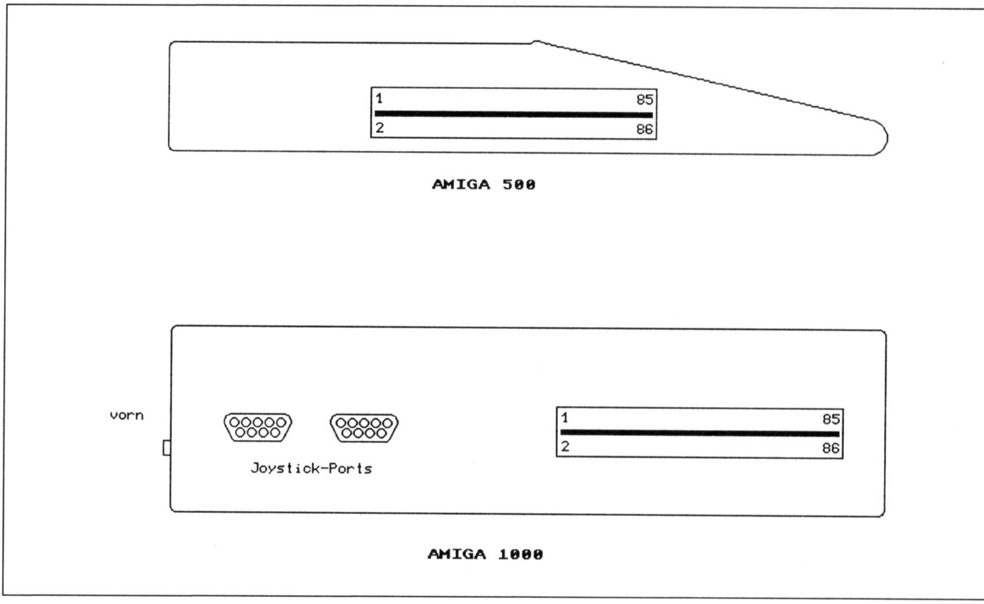


Bild 5.12: Anordnung der Bus-Stecker bei Amiga 500 und 1000

befindet sich jeweils links hinten auf der Leiterseite.

Ganz rechts ist der 86-polige MMU-Steckverbinder zu sehen. Er entspricht den Erweiterungssteckern beim Amiga 500 und 1000 und ist direkt mit der CPU und den übrigen Bausteinen verbunden. Damit eignet er sich vor allem für zeitkritische Anwendungen, wie beispielsweise schnelle Prozessorkarten oder eine Memory-Management-Unit (MMU). Die fünf 100-poligen Amiga-Slots dagegen sind über Treiber angeschlossen, um die Belastung der CPU zu begrenzen. Auch diese Slots entsprechen in ihrer Belegung den übrigen Steckverbindern, wurden jedoch um einige Leitungen erweitert, die meist nicht von Bedeutung sind. Normalerweise werden dort Zusatzkarten wie Speichererweiterungen oder I/O-Anwendungen zum Einsatz kommen. Die Zugriffe erfolgen dabei von der 68000-CPU auf der Hauptplatine aus. Die Steuerung der Treiber wurde jedoch so ausgelegt, daß auch Speicherzugriffe von den Slots her auf die Bausteine der Hauptplatine möglich sind. Die Adreßpegel werden dann von einer Slot-Karte – beispielsweise einer leistungsfähigen Prozessorkarte oder einem DMA-Controller – generiert und auf den internen Amiga-Bus durchgeschaltet. So läßt sich der Amiga beispielsweise von einer dort eingesteckten 68020-Karte betreiben. Alternativ können die Treiber mittels $\overline{\text{LOCAL OWN}}$ ganz gesperrt werden. In diesem Fall sind DMA-Operationen zwischen den Slots möglich, ohne den Amiga-Prozessor, bzw. in seinem Adreßraum ablaufende DMA-Zyklen zu stören. Dabei kann der Amiga seine volle Leistungsfähigkeit beweisen. Im Chip-RAM können unabhängig vom

Prozessor einmalig angestoßene Sound- oder Disk-Operationen ablaufen, während der Amiga-Prozessor im Fast-RAM mit voller Geschwindigkeit ein Programm abarbeitet. Gleichzeitig können Erweiterungskarten über das abgetrennte Slot-System Informationen austauschen bzw. ein dort befindlicher Prozessor kann davon unabhängig lokal arbeiten. Parallel dazu sind natürlich noch auf dem PC-Bus Aktivitäten einer eventuell eingesteckten PC- oder AT-Karte möglich!

Beim Amiga 1000 ist der 86-polige Expansion-Port an der rechten Gehäuseseite, beim Amiga 500 links jeweils hinter einer abnehmbaren Platikklappe verborgen. Bild 5.12 zeigt für beide Fälle die genaue Lage und die Pinbelegung. Als Stecker ist jeweils eine Platinenzunge mit den Kontakten vorgesehen, auf die entsprechende 86-polige Stecker passen. Leider ist es durch diesen Unterschied ohne einen speziellen Adapter nicht möglich, für den Amiga 500 oder 1000 konzipierte Karten direkt im MMU-Slot des Amiga 2000 zu betreiben bzw. umgekehrt.

Die im Amiga 2000 vorhandenen PC- und AT-Slots sind zunächst völlig getrennt von der übrigen Hardware. Sie sind lediglich untereinander verbunden und mit den dafür vorgesehenen Pins an die verschiedenen Betriebsspannungen +5 Volt, –5 Volt, +12 Volt und –12 Volt angeschlossen.

Aktiviert wird dieses Bussystem erst durch Einstecken einer PC- oder AT-Brückenkarte. Die Leitungen werden dann von der Karte aus mit den PC-Signalen beschaltet, so daß PC-Zusatzkarten benutzbar sind. Soll keine PC- oder AT-Brückenkarte zum Einsatz kommen, ist zum Betrieb von PC-I/O-Karten über den Amiga-Bus der in Kapitel 5.3.3.6 vorgestellte PC-Bus-Adapter empfehlenswert.

5.2.2 Pinbelegung der Bus-Steckverbinder

Tabelle 5.6 zeigt die Anschlußbelegung der fünf gepufferten 100-poligen Bus-Steckverbinder (Amiga-Slots) und Tabelle 5.7 die des 86-poligen MMU-Connectors im Amiga 2000. Die über einen Treiber geführten Signale führen in ihrer Bezeichnung zusätzlich ein B (buffered) vor dem Namen. Wo sich die Funktion der Signale von der bei anderen Amigas oder Slots unterscheidet, wurde eine Kennzeichnung (X) angebracht.

Die Expansionports des Amiga 500 und des Amiga 1000 entsprechen weitgehend dem MMU-Connector. Allerdings liegen am MMU-Connector nicht die Daisy-Chain-Leitungen für Autokonfiguration. Daher zeigt Tabelle 5.8 noch die genaue Belegung der Steckverbinder am Amiga 500 und 1000.

hinten

Signal	Pin			Pin	Signal
GND	1			2	GND
GND	3			4	GND
+5V	5			6	+5V
$\overline{\text{LOCAL OWN}}$	7	X		8	–5V
$\overline{\text{SLAVEn}}$	9	X		10	+12V
$\overline{\text{CONFIGOUTn}}$	11	X	X	12	$\overline{\text{CONFIGINn}}$
GND	13			14	$\overline{\text{C3B}}$
CDACB	15			16	$\overline{\text{C1B}}$
$\overline{\text{OVR}}$	17		X	18	XRDY
$\overline{\text{INT2}}$	19			20	–12V
BA5	21			22	$\overline{\text{INT6}}$
BA6	23			24	BA4
GND	25			26	BA3
BA2	27			28	BA7
BA1	29			30	BA8
BFC0	31			32	BA9
BFC1	33			34	BA10
BFC2	35			36	BA11
GND	37			38	BA12
BA13	39		X	40	$\overline{\text{EINT7}}$
BA14	41		X	42	$\overline{\text{EINT5}}$
BA15	43		X	44	$\overline{\text{EINT4}}$
BA16	45			46	$\overline{\text{BERR}}$
BA17	47			48	$\overline{\text{VPA}}$
GND	49			50	E
$\overline{\text{VMA}}$	51			52	BA18
$\overline{\text{RES}}$	53			54	BA19
$\overline{\text{HLT}}$	55			56	BA20
BA22	57			58	BA21
BA23	59			60	$\overline{\text{BRn}}$
GND	61			62	$\overline{\text{BGACK}}$
BD15	63			64	$\overline{\text{BGn}}$
BD14	65			66	$\overline{\text{DTACK}}$
BD13	67			68	READ
BD12	69			70	$\overline{\text{BLDS}}$
BD11	71			72	$\overline{\text{BUDS}}$
GND	73			74	$\overline{\text{BAS}}$
BD0	75			76	BD10

Bauteilseite

BD1	77			78	BD9
BD2	79			80	BD8
BD3	81			82	BD7
BD4	83			84	BD6
GND	85			86	BD5
GND	87			88	GND
GND	89			90	GND
GND	91		X	92	7ME
DOE	93	X	X	94	$\overline{\text{RESB}}$
$\overline{\text{BG}}$	95	X	X	96	$\overline{\text{EINT1}}$
RESERVED	97			98	RESERVED
GND	99			100	GND

Tabelle 5.6: Pinbelegung der Slots beim Amiga 2000. Die Angabe n bezieht sich auf die Slotnummer 1-5. Völlig andere Signale als beim MMU-Connector wurden mit X gekennzeichnet

hinten

GND	1			2	GND
GND	3			4	GND
+5V	5			6	+5V
NC	7	X		8	–5V
28M	9	X		10	+12V
NC	11	X	X	12	GND
GND	13			14	$\overline{\text{C3}}$
CDAC	15			16	$\overline{\text{C1}}$
$\overline{\text{OVR}}$	17		X	18	XRDY
$\overline{\text{INT2}}$	19			20	NC
A5	21			22	$\overline{\text{INT6}}$
A6	23			24	A4
GND	25			26	A3
A2	27			28	A7
A1	29			30	A8
FC0	31			32	A9
FC1	33			34	A10
FC2	35			36	A11

Signal	Pin		Pin	Signal	
GND	37		38	A12	
A13	39	X	40	$\overline{\text{IPL0}}$	
A14	41	X	42	$\overline{\text{IPL1}}$	
A15	43	X	44	$\overline{\text{IPL2}}$	
A16	45		46	$\overline{\text{BERR}}$	Bauteilseite
A17	47		48	$\overline{\text{VPA}}$	
GND	49		50	E	
$\overline{\text{VMA}}$	51		52	A18	
$\overline{\text{RES}}$	53		54	A19	
$\overline{\text{HLT}}$	55		56	A20	
A22	57		58	A21	
A23	59		60	$\overline{\text{BR}}$	
GND	61		62	$\overline{\text{BGACK}}$	
PD15	63		64	$\overline{\text{BG}}$	
PD14	65		66	$\overline{\text{DTACK}}$	
PD13	67		68	$\overline{\text{PRW}}$	
PD12	69		70	$\overline{\text{LDS}}$	
PD11	71		72	$\overline{\text{UDS}}$	
GND	73		74	$\overline{\text{AS}}$	
PD0	75		76	PD10	
PD1	77		78	PD9	
PD2	79		80	PD8	
PD3	81		82	PD7	
PD4	83		84	PD6	
GND	85		86	PD5	

Tabelle 5.7: Pinbelegung des MMU-Connectors beim Amiga 2000. Völlig andere Signale als bei den Amiga 2000 – Slots wurden mit X gekennzeichnet

	links				
GND	1			2	GND
GND	3			4	GND
+5V	5			6	+5V
EXP1	7	X		8	–5V
EXP2	9	X		10	+12V
EXP	11	X	X	12	CONFIG

	Signal	Pin		Pin	Signal
	GND	13		14	$\overline{\text{C3}}$
	CDAC	15		16	$\overline{\text{C1}}$
	$\overline{\text{OVR}}$	17	X	18	XRDY
	$\overline{\text{INT2}}$	19	X	20	$\overline{\text{PALOPE}}$
	A5	21		22	$\overline{\text{INT6}}$
	A6	23		24	A4
	GND	25		26	A3
	A2	27		28	A7
	A1	29		30	A8
	FC0	31		32	A9
	FC1	33		34	A10
	FC2	35		36	A11
	GND	37		38	A12
	A13	39		40	$\overline{\text{IPL0}}$
	A14	41		42	$\overline{\text{IPL1}}$
oben	A15	43		44	$\overline{\text{IPL2}}$
	A16	45		46	$\overline{\text{BERR}}$
	A17	47		48	$\overline{\text{VPA}}$
	GND	49		50	E
	$\overline{\text{VMA}}$	51		52	A18
	$\overline{\text{RES}}$	53		54	A19
	$\overline{\text{HLT}}$	55		56	A20
	A22	57		58	A21
	A23	59		60	$\overline{\text{BR}}$
	GND	61		62	$\overline{\text{BGACK}}$
	PD15	63		64	$\overline{\text{BG}}$
	PD14	65		66	$\overline{\text{DTACK}}$
	PD13	67		68	$\overline{\text{PRW}}$
	PD12	69		70	$\overline{\text{LDS}}$
	PD11	71		72	$\overline{\text{UDS}}$
	GND	73		74	$\overline{\text{AS}}$
	PD0	75		76	PD10
	PD1	77		78	PD9
	PD2	79		80	PD8
	PD3	81		82	PD7
	PD4	83		84	PD6
	GND	85		86	PD5

Tabelle 5.8: Pinbelegung des Expansionports beim Amiga 500 und 1000. Abweichungen gegenüber dem MMU-Connector des Amiga 2000 wurden mit X gekennzeichnet

5.2.3 Bus-Signale

Die Abläufe bei normalen Lese- und Schreibzugriffen auf Speicherstellen wurden bereits in Kapitel 5.1 erläutert. Außer den dort erwähnten Signalen gibt es im System aber noch eine Vielzahl anderer Steuerleitungen. Beim Amiga wurden die CPU-Signale unverändert an den Erweiterungssteckverbindern zugänglich gemacht. Für Zusatzschaltungen sind also die Original-Unterlagen zum 68000 von Motorola maßgeblich. Einzig bei der Behandlung von $\overline{DTACK}$ (Data Transfer Acknowledge) ergibt sich ein bedeutsamer Unterschied. Dieses asynchrone Signal dient dem Prozessor als Vorgabe, wann jeweils ein laufender Speicherzyklus abzuschließen ist. Es muß normalerweise von der adressierten Einheit generiert werden. Der Amiga übernimmt diese Aufgabe für den gesamten freien Speicher mit, so daß sich kürzestmögliche Zugriffszeiten ergeben. Das ist sinnvoll, da heute langsame Speicher ohnehin so gut wie nie zu finden sind. Daraus ergibt sich, daß $\overline{DTACK}$ an allen Erweiterungssteckverbindern des Amiga immer als Ausgang zu betrachten ist. Trotzdem kann bei Bedarf ein Speicherzyklus noch mit XRDY = LOW verzögert werden.

Bei der Verwendung mancher CPU-Signale ist also Vorsicht geboten. Genau wie das bereits erwähnte $\overline{DTACK}$ sollen die Ausgänge für synchrone Speicherzugriffe $\overline{VMA}$, $\overline{VPA}$ und E nicht benutzt werden. Das ist allerdings auch nicht sinnvoll, da solche Zugriffe beim 68000 vergleichsweise langsam ablaufen. Sie werden im Amiga zum Betrieb der Portbausteine 8520 benutzt.

Auch die Interrupt-Signale $\overline{IPL0}$..$\overline{IPL2}$ sind bereits vom System beschaltet. Im Spezialchip PAULA befindet sich ein programmierbarer Interruptcontroller, der auch die Decodierung übernimmt. Für externe Interrupts stehen die herausgeführten PAULA-Eingänge $\overline{INT2}$ und $\overline{INT6}$ zur Verfügung. Genauere Angaben enthält Kapitel 3.5.1. Die Bedeutung einiger spezieller Bus-Leitungen und ihre Verwendung beim Amiga soll hier kurz erläutert werden.

- FC0..FC3 (Function Code 0..3)

Die hier anliegende Bitkombination enthält den Status des Prozessors und kennzeichnet, auf welchen Daten- oder Programmbereich augenblicklich zugegriffen wird. Das ist wichtig für eine MMU (Memory-Management-Unit).

- $\overline{IPL0}$..$\overline{IPL2}$
  (Interrupt Priority Level 0..2)

Über diese Eingänge wird dem Prozessor eine Interrupt-Anforderung und deren Priorität mitgeteilt. Sind alle drei Eingänge HIGH, ist keine Interruptquelle aktiv.

- E

Taktsignal des 68000 für das Timing synchroner Speicherzugriffe bei langsamer Peripherie. (Normalerweise nicht benutzbar.)

- $\overline{OVR}$ (Override)

Mit diesem Signal kann die interne Generierung des $\overline{DTACK}$-Signals und die Adreßraum-Dekodierung außer Kraft gesetzt werden. Beim Amiga 500 und 2000 ist die Benutzung von $\overline{OVR}$ im Bereich \$200000 bis \$9FFFFF nicht möglich.

- $\overline{C1}$

Ist ein Taktsignal mit einer Frequenz von 3,58 MHz, synchronisiert auf die fallende Flanke des CPU-Takts von 7,16 MHz.

- $\overline{C3}$

Ist ebenso wie $\overline{C1}$ ein Taktsignal mit einer Frequenz von 3,58 MHz, jedoch synchronisiert auf die steigende Flanke des CPU-Takts von 7,16 MHz.

- CDAC

Taktsignal mit einer Frequenz von 7,16 MHz, das allerdings dem CPU-Takt um 90 Grad nacheilt.

Das Signal 7M des 68000-Prozessors wurde aufgrund seiner hohen Frequenz und der benötigten Genauigkeit nicht auf die Erweiterungs-Steckverbindungen herausgeführt. Es kann jedoch auf einfache Weise aus den Signalen $\overline{C1}$ und $\overline{C3}$ durch eine XNOR-Verknüpfung gewonnen werden. 7M ist HIGH, wenn $\overline{C1}$ und $\overline{C3}$ den gleichen Pegel haben. Bei Betrachtungen des Bus Timing auf den fünf Amiga-Slots müssen unbedingt die Verzögerungszeiten der zwischengeschalteten Treiber berücksichtigt werden.

Die im Zusammenhang mit der Konfiguration von Zusatzkarten wichtigen Signale $\overline{\text{CONFIGINn}}$ und $\overline{\text{CONFIGOUTn}}$ sowie $\overline{\text{SLAVEn}}$ werden im folgenden Kapitel eingehend erläutert.

5.2.4 Autokonfiguration

Commodore hat sich bei den Amiga-Computern ein spezielles Bus-Konzept für Erweiterungskarten ausgedacht. Es ermöglicht, daß bereits beim Einschalten des Systems von einer Autokonfigurations-Software erkannt wird, welche zusätzliche Einheiten angeschlossen sind. Das Betriebssystem kann diese dann im Adreßraum plazieren. Dazu erhalten alle Einheiten einen Autokonfigurationsteil und werden, wie Bild 5.13 zeigt, als Kette hintereinandergeschaltet.

Die Basisadresse aller angeschlossenen Zusätze beim Einschalten ist zunächst $E80000. Jeder Zusatz enthält einen separaten Select-Anschluß $\overline{\text{CONFIGINn}}$, der jeweils vom Anschluß $\overline{\text{CONFIGOUTn}}$ der vorgeschalteten Einheit gesteuert wird. Das Zeichen n steht dabei für die Slotnummer. Da alle Slots gleich aufgebaut sind, wird diese Nummer im Folgen-

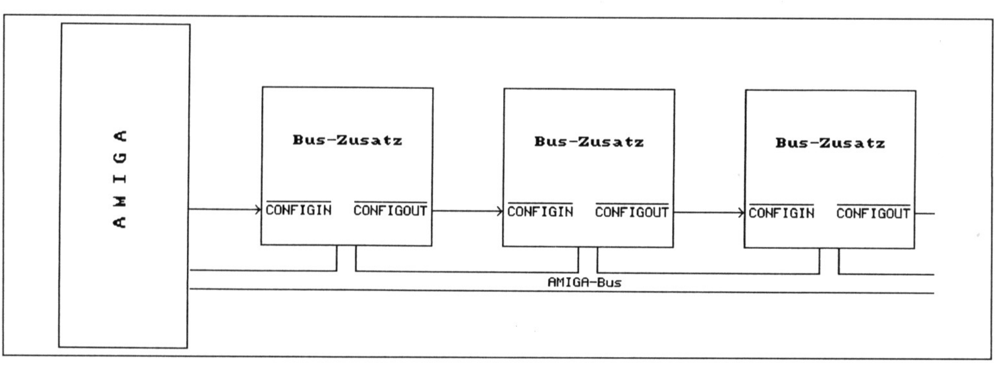


Bild 5.13: Die Autokonfigurationskette

den der Einfachheit halber fortgelassen. Nach dem Einschalten oder einem Reset führt der Anschluß $\overline{\text{CONFIGOUT}}$ auf allen Zusätzen HIGH-Pegel. Lediglich die erste Karte bekommt vom Amiga an $\overline{\text{CONFIGIN}}$ ein LOW-Signal. Wenn nun auf den Adreßbereich $E80000 zugegriffen wird, antwortet nur dieser erste Zusatz, weil ja alle anderen durch $\overline{\text{CONFIGIN}}$ = HIGH gesperrt sind.

Jeder Zusatz enthält eine Anzahl von speziellen Kenndaten, zum Beispiel Typ der Erweiterung und Größe des benötigten Adreßbereiches, die vom Prozessor ausgelesen werden können sowie ein Register, das mit einer neuen Basisadresse beschreibbar ist, und ein Flipflop zum Aktivieren von $\overline{\text{CONFIGOUT}}$. Die Autokonfigurations-Software fragt die Kenndaten ab, legt den Zusatz durch Beschreiben des Adreßregisters auf eine neue Basisadresse und setzt dabei dessen Ausgangssignal $\overline{\text{CONFIGOUT}}$ auf LOW, so daß die nachgeschaltete Erweiterung nun im Bereich $E80000 antworten kann.

So wird mit allen angeschlossenen Einheiten verfahren. Dabei entsteht eine Tabelle mit den Daten der Erweiterungen. Die Konfigurationssoftware kann die Zusätze je nach Art und Kombination automatisch optimal im Adreßraum plazieren, ohne daß sich Kollisionen ergeben. Die Basisadressen jeder Einheit werden vom Betriebssystem verwaltet. So entfällt das manuelle Einstellen von I/O-Adressen auf jeder Karte mit den sich dadurch ergebenden Fehlermöglichkeiten sowie das anschließende Patchen der Software.

Jede Zusatzkarte muß während sie Daten auf den Bus legt das Signal $\overline{\text{SLAVE}}$ nach LOW ziehen. Sollten trotz der Konfiguration bei einem Zugriff zwei Karten antworten, weil beispielsweise ein Hardwarefehler vorliegt, erkennt das der Amiga und bricht den Zugriff ab, bevor Schäden entstehen.

Die fünf 100-poligen Slots des Amiga 2000 bilden bereits eine Konfigurationskette, die jederzeit durch weitere Slots erweitert werden kann. Daher besitzen die Steckplätze jeweils einen Anschluß $\overline{\text{CONFIGIN}}$ und $\overline{\text{CONFIGOUT}}$. Eine zusätzliche Logik sorgt dafür, daß $\overline{\text{CONFIGOUT}}$ bei unbenutzten Slots trotzdem zum nächsten Steckplatz weitergeführt wird. Karten können also in beliebigen Slots stecken.

Amiga 500 und 1000 besitzen anstelle der Slots nur einen einzigen Bus-Stecker, doch auch dort findet sich ein Anschluß $\overline{\text{CONFIG}}$. Er kann genau wie beim Amiga 2000 an $\overline{\text{CONFIGIN}}$ einer angeschlossenen Erweiterung oder Erweiterungskette geführt werden. Das von dem Zusatz erzeugte Ausgangssignal $\overline{\text{CONFIGOUT}}$ wird dann an dem gleichen Pin für die nächste Einheit zugänglich gemacht.

Vom Betriebssystem Kickstart 1.1 wurde noch eine recht aufwendige Autokonfigurations-Hardware gefordert. Bereits Kickstart 1.2 unterstützt zusätzlich ein neues, wesentlich vereinfachtes Konzept, das vor allem billiger aufzubauen ist. Dabei wurde von der Realisierung mit einem PAL ausgegangen, was zwar technisch sehr elegant, im Hobbybereich aber leider kaum programmierbar und zudem (noch) teurer als Standard-Logik ist. Um mit kleinen PALs auszukommen, arbeitet das neue Konzept nibble-orientiert (Nibble =

Halbbyte = 4 Bit). Es werden jeweils die höchstwertigen vier Bit (D12..D15) benutzt. Weil oft acht Bit eine logische Einheit bilden, sind in der nachstehenden Tabelle jeweils zwei Nibble zu einer Zeile zusammengefaßt. Die angegebenen Adressen bilden den Offset von der Basisadresse $E80000 aus.

Weil die gängigen PALs in negativer Logik, das heißt mit invertierenden Ausgangstreibern gefertigt werden, sind die folgenden Kennwerte außer denen in Adresse $00/02 und $40/42 invertiert abzulegen.

Bei Zugriff auf die höherwertigen Byte-Adressen werden die Bits 0 bis 3 gelesen oder geschrieben, bei den niederwertigen die Bits 4 bis 7. Also läßt sich über Adresse $E80000 die Speichergröße und Chained Config Request lesen.

Adresse	R/W	Bedeutung	
$00/02	R	7 6 5 4 3 2 1 0	Boardtyp und -größe
		Bits 2, 1, 0	Speichergröße: 000 = 8 Megabyte 001 = 64 Kilobyte 010 = 128 Kilobyte 011 = 256 Kilobyte 100 = 512 Kilobyte 101 = 1 Megabyte 110 = 2 Megabyte 111 = 4 Megabyte
		Bit 3	1 = Chained Config Request (Das nächste Glied in der Config-Kette gehört physikalisch zu diesem)
		Bit 4	1 = optionaler Rom-Vektor in $28..2E ist gültig
		Bit 5	1 = zugewiesenen Adreßbereich als freien Speicher übernehmen
		Bits 7, 6	Board Typ: (00 = Board mit altem Config) (01 = Box mit altem Config) (10 = Box mit altem Config) (11 = Diese Spezifikation)

Adresse	R/W	Bedeutung
$04/06	R	Produkt Nummer Die Produktnummer (2 Byte) wird vom Hersteller vergeben. Sie wird von der Software benutzt, um Treiber zu initialisieren.
$08/0A	R	7 6 5 4 3 2 1 0 Bit 5–0: 00000 (Reserviert) Bit 6: 0 = Shut-up-Logik implementiert Bit 7: 1 = Board muß in den 8-Megabyte-Bereich
$0C/0E	R	Reserviert (Inhalt: $00)
$10/12	R	Mfg# (Herstellernummer) Highbyte
$14/16	R	Mfg# (Herstellernummer) Lowbyte Die Herstellernummer (2 Byte) wird von Commodore vergeben. Sie wird von der Software benutzt, um Treiber zu initialisieren.
$18/1A	R	Optionale Seriennummer Byte 0 (MSB)
$1C/1E	R	Optionale Seriennummer Byte 1
$20/22	R	Optionale Seriennummer Byte 2
$24/26	R	Optionale Seriennummer Byte 3 (LSB) Die Seriennummer (4 Byte) wird vom Hersteller vergeben und erscheint in der Autokonfigurationstabelle.
$28/2A	R	Optionaler ROM-Vektor (Highbyte)
$2C/2E	R	Optionaler ROM-Vektor (Lowbyte) Dieser 2-Byte-Vektor zeigt auf eine Initialisierungsstruktur, in der zwei Vektoren stehen, von denen der erste (DiagPoint) benutzt wird, um ein Diagnose-Programm anzuspringen, das überprüft, ob das Board o.k. ist bzw. die Treiberprogramme relociert. Über den zweiten Vektor (BootPoint) wird eventuell gebootet. Wenn Bit 4 im Nibble 0 gesetzt ist, muß mindestens der erste Vektor auf ein korrektes Programm zeigen. Diese Angaben gelten ab Kickstart-Version 1.3 aufwärts. Unter Kickstart 1.2 hängt sich das System auf.
$30/32	R	Reserviert (Inhalt: 0)
	W	Optional: Schreibzugriff setzt das Adreßregister zurück

Adresse	R/W	Bedeutung		
$34/36	R	Reserviert (Inhalt: 0)		
$38/3A	R	Reserviert (Inhalt: 0)		
$3C/3E	R	Reserviert (Inhalt: 0)		
$40/42	R/W	7 6 5 4 3 2 1 0 optionales Kontroll/Statusregister		
		Bit	Schreiben	Lesen
		0	Interrupt enable	Interrupt enable
		1	undefiniert	undefiniert
		2	lokaler Reset	muß 0 sein
		3	undefiniert	undefiniert
		4	undefiniert	INT2 aktiv
		5	undefiniert	INT6 aktiv
		6	undefiniert	INT7 aktiv
		7	undefiniert	ein INT aktiv
$44/46	R	Reserviert (Inhalt: 0)		
$48/4A	W	Basisadreßregister Diese Bits bilden die Basisadresse des Boards und werden mit A23..A16 verglichen.		
$4C/4E	W	Optionale »ShutUp«-Adresse Ein Schreibzugriff auf $4C veranlaßt das Board $\overline{\text{CONFIGOUT}}$ aus- zugeben und dann auf keinen Zugriff mehr zu antworten.		
$50/52	R	Reserviert (Inhalt: 0)		
$54/56	R	Reserviert (Inhalt: 0)		
$58/5A	R	Reserviert (Inhalt: 0)		
$5C/5E	R	Reserviert (Inhalt: 0)		
$60/62	R	Reserviert (Inhalt: 0)		
$64/66	R	Reserviert (Inhalt: 0)		
$68/6A	R	Reserviert (Inhalt: 0)		
$6C/6E	R	Reserviert (Inhalt: 0)		
$70/72	R	Reserviert (Inhalt: 0)		
$74/76	R	Reserviert (Inhalt: 0)		
$78/7A	R	Reserviert (Inhalt: 0)		
$7C/7E	R	Reserviert (Inhalt: 0)		

5.2.5 Der Boot-Vorgang

Die von Commodore lange Zeit zu ihren Amiga-Computern gelieferte Betriebssystemversion Kickstart 1.2 beherrscht noch nicht das Booten von einer Erweiterungskarte. Das System muß immer von einer Diskette aus hochgefahren werden. Die im folgenden gemachten Angaben beziehen sich daher auf Kickstart-Versionen ab 1.3 aufwärts.

Nach dem Einschalten des Computers oder einem Reset wird die EXEC-Library aktiviert. EXEC sucht im Speicher nach Resident-Strukturen und initialisiert sie. Dabei werden auch verschiedene andere Libraries ins System geholt, z.B. Graphics und Intuition. Außerdem werden Devices ins System eingebunden, z.B. Input- und Trackdisk-Device. Als letzte Library wird Expansion-Lib eingebunden und die darin enthaltene Funktion ConfigChain aufgerufen. Sie liest unter $E80000 die Information über das erste Board in der Konfigurationskette, konfiguriert es, indem sie es auf eine andere Basisadresse legt und im System eine Struktur erzeugt, die alle wichtigen Daten über das Board enthält.

Wenn durch ein gesetztes Bit 4 in Nibble 0 des Config-Bereichs der optionale ROM-Vektor aktiviert wurde, kopiert der Amiga auch die Initialisierungsstruktur mit dem zugehörigen Datenbereich ins RAM. Über den DiagPoint wird ein Diagnose-Programm zur Überprüfung des Boards aufgerufen, das beispielsweise eine Checksumme kontrolliert und die Treibersoftware durch Eintragen der Boardadresse relociert.

Dies wird für jedes Board getan und Schritt für Schritt eine »BoardList« aufgebaut, in der alle Strukturen (ConfigDef-Nodes) enthalten sind, die bis dahin erzeugt wurden.

EXEC baut das System nun weiter auf, bis nur noch die DOS-Library fehlt. An dieser Stelle wird unter Kickstart 1.3 eine ROM-Boot-Library aufgerufen. Sie durchsucht für jedes Board aus der BoardList den zugehörigen Speicher nach einer Resident-Struktur und initialisiert diese gegebenenfalls. Dabei wird ein Programmstück ausgeführt, das zwei Aufgaben hat: Einmal wird eine DeviceNode erzeugt, in der alle zur Device gehörigen Daten verzeichnet sind, zum zweiten eine BootNode, in der ein Zeiger auf die DeviceNode steht und eine Boot-Priorität angegeben wird. Diese BootNode wird schließlich über die EXEC-Funktion Enqueue in die BootList eingetragen. Dabei wird die Priorität berücksichtigt.

Nun wird eine Funktion der Strap-Library aufgerufen. Sie überprüft, ob im Laufwerk DF0: eine bootfähige Diskette eingelegt ist. Falls ja, wird diese Diskette unwiederruflich gebootet, das heißt, die beiden Bootblöcke werden geladen und der darin enthaltene Code aufgerufen. Dabei wird üblicherweise die DOS-Library aktiviert. Andernfalls geht die Strap-Library-Funktion durch die BootList und ruft für jede darin enthaltene BootNode den zum Board gehörenden BootPoint auf. Wenn das darüber aufgerufene Programmstück tatsächlich bootet, das heißt die DOS-Library aktiviert, kehrt der Aufruf nicht wieder zurück.

5.3 Die RAM/ROM-Karte

Diesem Buch liegt die Leerplatine einer autokonfigurierenden, bootfähigen Speicherkarte bei, die vielseitig einsetzbar ist. Neben einem gemischten Betrieb als gleichzeitige ROM-Disk, reset- und ausschaltfeste CMOS-RAM-Disk und Speichererweiterung ist über einen Adapter auch der Anschluß diverser PC-Zusatzgeräte möglich. Die Autoren betrieben von dieser Karte aus eine Hard-Disk mit OMTI-Controller, so daß nach dem Einschalten ohne Einsatz einer Boot-Diskette das System schnell vollständig hochgefahren wird.

5.3.1 Schaltung der Karte

Die Spezifikation der in Abschnitt 5.2.2 beschriebenen Autokonfigurations-Logik wurde zwar im Hinblick auf eine PAL-Realisation entworfen. Trotzdem lassen sich die Anforderungen auch mit relativ einfachen Mitteln allein durch TTL-Standardbausteine und mindestens ein EPROM erfüllen, so daß der Aufbau im Hobby-Bereich ohne Spezialgeräte wie etwa einem PAL-Programmierer möglich ist. Der Beweis dafür ist die Schaltung der RAM/ROM-Karte. Sie ist mit einer vollständigen Autokonfigurationslogik ausgestattet und ermöglicht unter anderem auch das Booten des Systems. Flipflops, Adreßregister und Komparator sind bei der PAL-Realisation ebenfalls nötig und ein EPROM zur Aufnahme von Treiber- bzw. Testsoftware ist auf Zusätzen ohnehin wünschenswert, bei autobootenden Karten sogar unabdingbar nötig, so daß der Aufwand für die Autokonfiguration selbst relativ gering bleibt.

5.3.1.1 Festlegung der Kartengröße

Je nach benötigtem Adreßraum der Slot-Karte müssen die logischen Zustände bestimmter Adreßleitungen für die Selektierung ausgewertet werden. Für Autokonfiguration sind Größen von 64 Kilobyte bis 8 Megabyte vorgesehen. Auf der RAM/ROM-Karte sollten EPROMs bzw. CMOS-RAMs Verwendung finden, wovon auf der zur Verfügung stehenden Platinenfläche acht Speicher-Sockelpaare nebeneinander untergebracht werden

Adreßbit :	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
bei 64K:	1	1	1	0	1	0	0	0	S	S	S	X	X	X	X	X
bei 128K:	1	1	1	0	1	0	0	S	S	S	X	X	X	X	X	X
bei 256K:	1	1	1	0	1	0	S	S	S	X	X	X	X	X	X	X
bei 512K:	1	1	1	0	1	S	S	S	X	X	X	X	X	X	X	X
bei 1M:	1	1	1	0	S	S	S	X	X	X	X	X	X	X	X	X
bei 2M:	1	1	1	S	S	S	X	X	X	X	X	X	X	X	X	X
bei 4M:	1	1	S	S	S	X	X	X	X	X	X	X	X	X	X	X
bei 8M:	1	S	S	S	X	X	X	X	X	X	X	X	X	X	X	X

Tabelle 5.9: Mögliche Zuordnung der Select-Signale (S) bei verschiedenen Board-Größen (Pegel für Konfigurations-Adreßraum)

Bank 8x	Select-Ein			Bank 1	Bank 2	Bank 3	...	Bank 8
8K	A13	A14	A15	E80000	E82000	E84000	...	E8E000
16K	A14	A15	A16	E80000	E84000	E88000	...	E9C000
32K	A15	A16	A17	E80000	E88000	E90000	...	EB8000
64K	A16	A17	A18	E80000	E90000	EA0000	...	EF0000
128K	A17	A18	A19	E00000	E20000	E40000	...	EE0000
256K	A18	A19	A20	E00000	E40000	E80000	...	FC0000
512K	A19	A20	A21	C00000	C80000	D00000	...	F80000
1M	A20	A21	A22	800000	900000	A00000	...	F00000

Tabelle 5.10: Basisadressen der Sockel bei verschiedenen Speichergrößen pro Bereich

konnten. Für die Selektierung werden also drei Adreßleitungen benötigt. Tabelle 5.9 gibt einen Überblick, welche Bedeutung die einzelnen Adreßbits bei den verschiedenen Speicherbereichen des Boards dann haben.

S steht für Anschluß an einem Select-Eingang des Demultiplexers. Der Zustand der höherwertigen Adreßleitungen bestimmt die Basisadresse des Zusatzes. In Tabelle 5.9 wurden die Pegel im unkonfigurierten Zustand angegeben. Sie müssen bei der Konfiguration softwaremäßig verändert werden. X kennzeichnet beliebige Pegel zum Adressieren des Boards selbst.

Tabelle 5.10 zeigt die Adreßzuordnung für die einzelnen Sockel bei den sich ergebenden Speichergrößen pro Bereich im Konfigurationsadreßraum. Weil die größten zur Zeit relativ preisgünstig beschaffbaren Speicherbausteine 64 Kilobyte mit acht Bit Wortbreite fassen (EPROM 27512), also 128 Kilobyte bei 16 Bit Wortbreite (2 Stück), wurde die in den Tabellen hervorgehobene Zuordnung für die Speicherkarte ausgewählt.

5.3.1.2 TTL-Konfigurationslogik

Bild 5.14 zeigt den Schaltplan der RAM/ROM-Platine. Zentraler Teil der gesamten Selektierungslogik ist der 4-Bit-Komparator U26. Er gibt am Ausgang A=B nur dann HIGH ab, wenn die beiden Eingangskombinationen an A und B genau gleich sind und zusätzlich auch der Übertrags-Eingang O HIGH ist sowie $\overline{\text{E1}}$ und $\overline{\text{E2}}$ LOW. Die rechts oben abgebildeten beiden Flipflops im IC U23 werden durch den Reset-Impuls des Amiga in einen definierten Anfangszustand gebracht. Damit liegt SHUTUP auf LOW und $\overline{\text{CONFIGOUT}}$ auf HIGH. Die Ausgänge des Registers U25 sind wegen HIGH an $\overline{\text{OE2}}$ also hochohmig.

Der Eingang O des Komparators U26 ist über ein NOR-Gatter beschaltet, das $\overline{\text{CONFIGIN}}$ mit SHUTUP verknüpft. Solange $\overline{\text{CONFIGIN}}$ HIGH führt, liegt BOARDSEL auf LOW und die Erweiterung bleibt inaktiv. Die Eingänge A kommen vom 4-Bit-Latch U25, das hier die Funktion des Basisadreßregisters erfüllt. Da sein Reset-Eingang nicht ausgenutzt

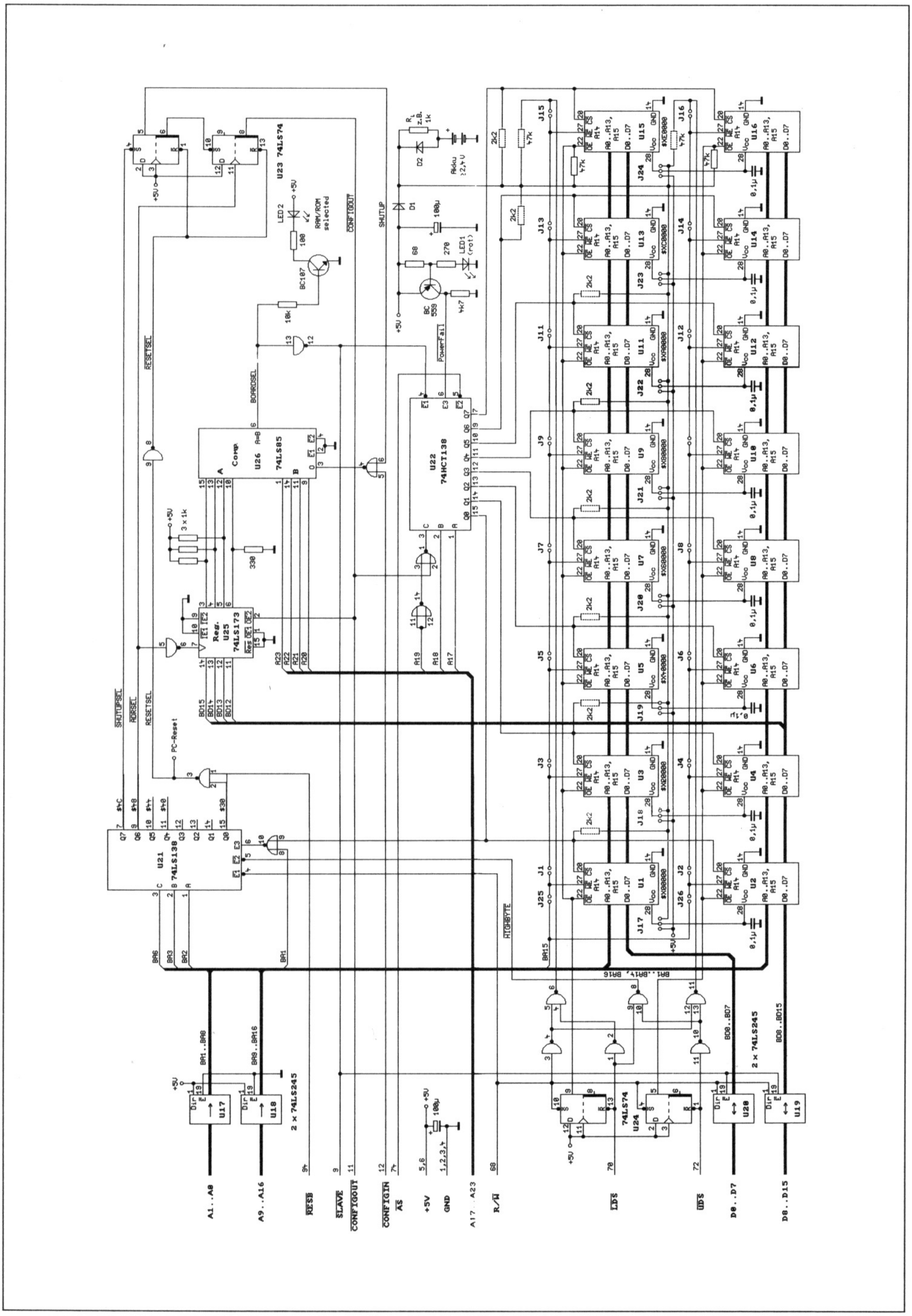


Bild 5.14: Schaltplan der RAM/ROM-Karte

wird, ist der Inhalt nach dem Einschalten zwar zufällig, was aber unerheblich ist, da seine Ausgänge zunächst wegen HIGH an $\overline{\text{OE2}}$ einen hochohmigen Zustand besitzen. Der Pegel der Komparator-Eingänge A wird also allein von den vier Widerständen bestimmt, die so ausgewählt wurden, Nach dem Auslesen der Parameter wird der Prozessor den Wert der neuen Basisadresse nach \$E80048 ausgeben. Sehen Sie sich zum Ablauf dieser Operation Bild 5.2 (das Zeitdiagramm des 68000-Schreibzyklus) an. Die Adresse \$E80048 entspricht binär:

Adreßbit:	23	22	21	20	19	18	17	16	...	7	6	5	4	3	2	1	0
Wert:	1	1	1	0	1	0	0	0	...	0	1	0	0	1	0	0	0
hexadezimal:		E					8		...		4				8		

daß sich an A der Wert \$E ergibt: der höchstwertige Teil der Basisadresse des Config-Bereiches.

Sobald jetzt $\overline{\text{CONFIGIN}}$ LOW wird, und eine Adresse im Bereich \$EXXXXX anliegt, wird BOARDSEL HIGH. Das aktiviert den Demultiplexer U22 und hat ein LOW an demjenigen seiner Ausgänge Q0 bis Q7 zur Folge, dessen Nummer der Kombination an den Select-Eingängen A, B und C entspricht. Solange $\overline{\text{CONFIGOUT}}$ noch HIGH führt, bleibt C wegen der beiden NOR-Gatter immer LOW. Im Bereich der Autoconfig-Adressen (\$E800XX) werden also trotz des gesetzten Adreßbits A19 immer die EPROMs in U1 und U2 aktiviert. Dort müssen die im vorigen Abschnitt spezifizierten Informationen über die Erweiterung abgelegt sein. Gleichzeitig wird für Schreibzugriffe (R/$\overline{\text{W}}$ = LOW) auf High-Bytes ($\overline{\text{HIGHBYTE}}$ = LOW) mit geraden Wortadressen (BA1 = LOW) der Demultiplexer U21 aktiviert. Damit wird das Beschreiben des Registers bzw. das Setzen des Config- und ShutUp-Flipflops ermöglicht.

Beim Anlegen dieser Bitkombination wird BOARDSEL aufgrund der vier höchstwertigen Adreßbits sofort auf HIGH gehen. Wegen A18, A17 und A16 = LOW geht daraufhin Q0 von U22 nach LOW. Sobald wenig später auch noch allein $\overline{\text{UDS}}$ LOW wird (Bytezugriff), sind alle Bedingungen zum Durchschalten des Demultiplexers U21 erfüllt ($\overline{\text{E1}}$, E2 = LOW, E3 = HIGH), und der dem Eingangsvektor an A2, A3 und A6 zugeordnete Ausgang Q6 ($\overline{\text{ADRSEL}}$) geht ebenfalls nach LOW. Die Verknüpfung von $\overline{\text{LDS}}$ und $\overline{\text{UDS}}$ bewirkt, daß nur Bytezugriffe möglich sind, die Einbeziehung von A1 verhindert, daß beim Zugriff auf die Config-Adresse \$4A auch das Register unter \$48 beeinflußt wird. An $\overline{\text{ADRSEL}}$ ist über einen Inverter der CLOCK-Eingang des Registers U25 angeschlossen. Mit der dort entstehenden steigenden CLOCK-Flanke übernehmen die im Registerbaustein enthaltenen D-Flipflops die anstehenden Daten. Sie erscheinen aber noch nicht an den Ausgängen, weil $\overline{\text{OE2}}$ noch HIGH führt. Nach einiger Zeit geht am Ende des Zu-

griffs UDS wieder nach HIGH, wodurch HIGHBYTE und ADRSEL HIGH werden. Jetzt fühlt sich das ebenfalls an ADRSEL angeschlossene positiv-flankengetriggerte D-Flipflop im Baustein U23 angesprochen.

Es ändert seinen Zustand und setzt CONFIGOUT auf LOW. Gleichzeitig schaltet dieses Signal über OE2 die Ausgänge des Registers U25 auf den Komparator U26 durch.

Die Widerstände haben nun keinen Einfluß mehr und die neue Basisadresse ist für alle folgenden Zugriffe maßgeblich.

Bei jedem Zugriff auf die Schaltung zieht BOARDSEL auch den Ausgang SLAVE nach LOW. Dies ist wichtig, um Adreßkonflikte zu erkennen. Leider verfügen nämlich nicht alle Zusätze für den Amiga über eine Autoconfig-Logik. Diese Zusätze sind immer in einem festen Adreßbereich aktiv. Es könnte also vorkommen, daß zwei Karten auf einen Lesezugriff antworten. Die Folge wären falsche Pegel auf den Datenleitungen, schlimmstenfalls sogar bleibende Defekte in der Hardware.

Der Amiga 2000 verfügt daher über eine Logik, die erkennt, ob gleichzeitig mehr als zwei Einheiten aktiv sind. In einem solchen Fall wird das BERR-Signal aktiviert, was von der Software ausgewertet werden kann. Damit diese Vorrichtung funktioniert, muß selbstverständlich jeder Zusatz die Leitung SLAVE ordnungsgemäß bedienen.

Über das ShutUp-Flipflop läßt sich die gesamte Karte bis zum nächsten Reset deaktivieren. Dies geschieht mit einem Schreibzugriff auf die Offset-Adresse $4C.

Die Leitung SHUTUPSEL wird dabei kurz LOW, was das Flipflop im Baustein U23 (im Schaltplan rechts oben) setzt und über SHUTUP = HIGH ein HIGH auf BOARDSEL dauerhaft verhindert.

Zugriffe auf die Autokonfigurationsregister und Flipflops sind nur bei Schreibzugriffen auf gerade Adressen im Bereich der Sockel U1 und U2 möglich, weil der Demultiplexer U21 mit dem entsprechenden Select-Signal freigegeben wird.

Da sich in diesen Sockeln ROM-Bausteine bzw. speziell genutzte RAMs befinden (siehe Kapitel 5.3.3.5), kommen im konfigurierten Betrieb solche Zugriffe nicht mehr vor. Eine unbeabsichtigte Änderung der Zustände ist also bei Nutzung gemäß dieser Anleitung nicht zu befürchten.

5.3.1.3 Ansprechen der Speicherbausteine

Alle an den RAM/ROM-Sockeln verwendeten Adreß- und Datenleitungen werden über Treiber geführt, um die Belastung des Prozessorbusses zu verringern. Für die Adreßleitungen A1..A16 sind die beiden Bausteine U17 und U18 zuständig, für die Datenleitungen D0..D15 die Bausteine U19 und U20.

Den Namen der gepufferten Signale wurde ein B (buffered) vorangestellt. Die Pinzuordnung der Signale an den Treiberbausteinen ist nicht im Schaltplan enthalten und wird daher in den folgenden Tabellen zusammengefaßt.

Signal	Slot-Pin	vom Slot	von Karte
A1	29	3	17
A2	27	5	15
A3	26	6	14
A4	24	8	12
A5	21	9	11
A6	23	7	13
A7	28	4	16
A8	30	2	18

Tabelle 5.11: Pinzuordnung der Signale am Treiberbaustein U18

Signal	Slot-Pin	vom Slot	von Karte
A9	32	9	11
A10	34	8	12
A11	36	7	13
A12	38	6	14
A13	39	5	15
A14	41	4	16
A15	43	3	17
A16	45	2	18

Tabelle 5.12: Pinzuordnung der Signale am Treiberbaustein U17

Signal	Slot-Pin	vom Slot	von Karte
D0	75	15	5
D1	77	17	3
D2	79	16	4
D3	81	18	2
D4	83	11	9
D5	86	12	8
D6	84	13	7
D7	82	14	6

Tabelle 5.13: Pinzuordnung der Signale am Treiberbaustein U20

Signal	Slot-Pin	vom Slot	von Karte
D8	80	16	4
D9	78	11	9
D10	76	12	8
D11	71	13	7
D12	69	14	6
D13	67	15	5
D14	65	18	2
D15	63	17	3

Tabelle 5.14: Pinzuordnung der Signale am Treiberbaustein U19

Für alle Treiber wurden aus layouttechnischen Gründen bidirektionale Ausführungen vom Typ 74LS245 vorgesehen. Bei den Adreßleitungen besteht mit DIR = HIGH und $\overline{\text{E}}$ = LOW immer die Durchschaltung vom Slot zu den Sockeln. Die Datenleitungen müssen jedoch in ihrer Richtung umgeschaltet werden und dürfen nicht immer durchgeschaltet bleiben. Die Umschaltung der Richtung erfolgt je nach Schreib- oder Lesezyklus mit dem Signal R/$\overline{\text{W}}$. Eine Aktivierung der Treiber erfolgt immer dann, wenn das Board selektiert ist.

Die Zugriffe auf die Speicherbänke laufen nach den Zeitdiagrammen in Bild 5.2 (Schreiben) und 5.3 (Lesen) ab. Zu Beginn werden also vom Prozessor die Adreßpegel auf A0 bis A23 gelegt. Repräsentiert die Kombination eine Adresse innerhalb des Bereichs der Karte (Zustand auf A20 bis A23 entsprechend den Ausgangspegeln des Registers U25), geht BOARDSEL nach HIGH. Dies wird durch Aufleuchten der über einen Transistor angesteuerten LED2 (SelLED) angezeigt. Sobald zusätzlich $\overline{\text{AS}}$ LOW wird, bestimmen A17 bis A19, welcher Ausgang des Demultiplexers U22 LOW wird und damit über $\overline{\text{CS}}$ eine Speicherbank selektiert.

Handelt es sich um einen Lesebefehl, führt die Leitung R/$\overline{\text{W}}$ HIGH-Pegel. Das LOW hinter dem nachgeschalteten Negierer sorgt also für durchgehendes HIGH an den beiden im Schaltplan wie auf der Platine ganz rechts angeordneten 2-Pin-Jumpern J15 und J16 sowie am Anschluß $\overline{\text{WE}}$ aller mit gesteckten Jumpern damit verbundener Sockel. Dies ist für RAMs interessant. An ROMs liegt von J25, J26 her und über J1, J2 sowie weitere gesteckte Jumper der Zustand von A16 des Prozessors. Mit der fallenden Flanke von $\overline{\text{LDS}}$ und/oder $\overline{\text{UDS}}$ werden je nach Zugriffsbereich (Lowbyte, Highbyte oder Wort) die Flipflops zurückgesetzt, wodurch der Anschluß $\overline{\text{OE}}$ aller angeschlossenen Sockel sicher LOW wird und die über LOW an $\overline{\text{CS}}$ selektierten Bausteine den Inhalt der adressierten Speicherstelle auf den Datenbus legen.

Bei einem Schreibbefehl liegt auf der Leitung R/$\overline{\text{W}}$ LOW-Pegel. Dadurch werden beide angeschlossenen Flipflops im IC U24 gesetzt. Der Anschluß $\overline{\text{OE}}$ aller Sockel bleibt also während des gesamten

Zyklus fest auf HIGH. Auch LOW an $\overline{\text{LDS}}$ oder $\overline{\text{UDS}}$ ändern daran nichts. Über die Negierer wird der Zustand von R/$\overline{\text{W}}$ an den Ausgang der NAND-Gatter und damit an den Anschluß $\overline{\text{WE}}$ der angeschlossenen RAMs durchgeschaltet. Mit der ansteigenden Flanke dort übernehmen die mit LOW an $\overline{\text{CS}}$ selektierten RAMs die Informationen vom Datenbus in die adressierte Speicherstelle.

Am Prozessor MC68000 existiert wegen der 16-Bit-Architektur keine Leitung A0. Bei den RAMs bzw. ROMs wird daher A0 mit BA1 vom Prozessor verbunden, A1 mit BA2, A2 mit BA3 und so fort. Die Pinbelegung der verwendbaren Speicher-ICs ist in Anhang A enthalten. Damit ohne Änderungen wahlweise RAM- oder ROM-Bausteine einsetzbar sind und ein zusammenhängender Speicherbereich in den RAMs entsteht, wurde BA15 vom Prozessor an Pin 1 der Sockel (A14) gelegt. Damit sind RAM-Bausteine vom Typ 62256 korrekt belegt. Bei ROMs sollte an Pin 1 (A15) die Adreßleitung BA16 vom Prozessor ankommen. Stattdessen wird sie hier über Jumper an Pin 27 (A14) geführt. A14 und A15 sind bei ROMs also vertauscht. Bei EPROMs vom Typ 27512 ergibt sich die in Tabelle 5.15 angegebene Reihenfolge der vier Bereiche.

A15	A14	Bereich
0	0	\$0000 ... \$3FFF
1	0	\$8000 ... \$BFFF
0	1	\$4000 ... \$7FFF
1	1	\$C000 ... \$FFFF

Tabelle 5.15: Die Reihenfolge der sich durch Vertauschung von A14 und A15 ergebenden Adreßbereiche im EPROM 27512

Diese Aufteilung muß bei der Programmierung der Bausteine berücksichtigt werden! Das in Kapitel 5.3.3.1 vorgestellte Programm MakeRomDisk zum Zusammenstellen des EPROM-Inhalts gleicht den Effekt jedoch aus.

5.3.1.4 Datenpufferung über Akku

Damit die in CMOS-RAMs gespeicherten Informationen nach dem Ausschalten des Rechners nicht verlorengehen, muß ihre Spannungsversorgung gepuffert werden. Dazu ist eine Spannung von mindestens 2,0 Volt an den Bausteinen nötig. Allerdings beträgt die Stromaufnahme in dieser Betriebsart nur wenige Mikroampere, so daß ein Akku die Informationen durchaus ein Jahr lang speichern kann, ohne daß er zwischendurch aufgeladen werden müßte. Allerdings ist das nur der Fall, wenn sich die Steueranschlüsse $\overline{\text{OE}}$, $\overline{\text{WE}}$ und $\overline{\text{CS}}$ relativ zur Versorgungsspannung auf HIGH-Potential befinden. Daher wurden diese Leitungen mit Pull-up-Widerständen beschaltet.

Im Schaltplan ist der Akku und seine Ladeschaltung ganz rechts zu erkennen. Bei ausgeschalteter Betriebsspannung sperrt die Diode D1. Damit kann der Strom aus dem Akku über D2 nur zu den entsprechend gejumperten Sockelpaaren abfließen. Bei eingeschalteter Betriebsspannung sperrt D2. Die Stromversorgung der RAMs erfolgt jetzt über D1 vom Netzteil. Gleichzeitig wird der Akku über den parallel zu D2 liegenden Widerstand R_L geladen. Mit Hilfe der 3-Pin-Jumper J17..J24 kann für jedes RAM/ROM-Sockelpaar gewählt werden, ob die Be-

triebsspannung vom Netzteil (Jumperposition links) oder vom Akku (Jumperposition rechts) zugeführt werden soll.

Damit sich die Informationen in den RAMs nicht beim Ausschalten durch zufällige Effekte verändern, muß während des Absinkens der Betriebsspannung jeder Zugriff bereits frühzeitig gesperrt werden. Das gewährleistet die Transistorschaltung am high-aktiven Enable-Eingang E1 des Demultiplexers U22, eventuell zusammen mit den Pull-up-Widerständen am Anschluß $\overline{CS}$ jedes Sockels. Die Basis des PNP-Transistors liegt an einem Spannungsteiler. LED1 dient dazu, den Arbeitsbereich des Transistors zu verbessern. Sie ist zur Funktion der Schaltung wichtig und darf nicht fortgelassen werden. Der Transistor wird bei voller Betriebsspannung aufgesteuert und zieht den Demultiplexer-Eingang $\overline{E2}$ auf HIGH. Zugriffe auf die Speicherbausteine ($\overline{CS}$ = LOW) sind damit möglich.

Sinkt die Betriebsspannung jedoch unter etwa 4,5 Volt ab, sperrt der Transistor und legt LOW an den Demultiplexer. Die Selektierung der RAM/ROM-Sockel wird definitiv gesperrt.

Dadurch sind keinerlei zufällige Änderungen des RAM-Inhalts möglich. Die im Schaltplan angedeuteten Pull-up-Widerstände sind nur nötig, wenn für den Demultiplexer U22 eine andere Ausführung als der CMOS-Baustein 74HCT138 gewählt wird. Beim weiteren Absinken der Spannung halten die Pull-up-Widerstände die $\overline{CS}$-Leitungen dann sicher auf HIGH. Allerdings stellt sich ein höherer Akku-Strom ein.

5.3.2 Aufbau und Bestückung der Platine

Mit der diesem Buch beiliegenden doppelseitig ausgeführten und bereits durchkontaktierten Platine ist ein problemloser Aufbau der Schaltung gewährleistet. Trotzdem sollte mit Sorgfalt gearbeitet werden. Richten Sie sich nach den allgemeinen Bestückungs-Hinweisen in Anhang A und arbeiten Sie zusätzlich vor dem Bestücken die folgenden Bemerkungen genau durch.

5.3.2.1 Der Grundausbau

Die Bestückung erfolgt nach dem Lageplan in Bild 5.15. Alle benötigten Bauteile enthält Tabelle 5.16. Vergessen Sie nicht die drei Lötbrücken direkt über den Steckverbindungen. Wenigstens für die 16 RAM/ROM-Plätze U1..U16 sind unbedingt Stecksockel vorzusehen, nach Möglichkeit Präzisionsausführungen mit gedrehten Pins, um Kontaktprobleme auch bei vielen Bauteilwechseln zu vermeiden. Beachten Sie besonders die Einbaulage der ICs, Dioden-Transistoren und Elkos. Die beiden auf der Platine links oben vorgesehenen Widerstände bleiben im Normalfall unbestückt.

Nach der Fertigstellung kontrollieren Sie das Board unbedingt nocheinmal intensiv anhand des Schaltplanes auf kalte oder fehlende Lötstellen sowie versehentlich entstandene Lötbrücken und Kurzschlüsse! Die Jumper bestehen aus entsprechend abgetrennten einreihigen Pfostensteckerleisten, auf die Kurzschlußbrücken gesteckt werden können. Ebenso ist die Steckverbindung für die Select-Anzeige

1	Platine (liegt dem Buch bei)
5	IC-Sockel 14-pol
4	IC-Sockel 16-pol
4	IC-Sockel 20-pol
16	Sockel, 28-pol, Bestückung laut Text (U1..U16)
4	Bidirektionale Treiber 74LS245 (U17..U20)
1	Demultiplexer 74LS138 (U21)
1	Demultiplexer 74HCT138 (U22)
2	D-Flipflops 74LS74 (U23, U24)
1	4-Bit-Register 74LS173 (U25)
1	Komparator 74LS85 (U26)
1	Vierfach-NAND 74LS00
1	Vierfach-NOR 74LS02
1	Sechsfach-NOT 74LS04
1	Transistor BC107
1	Transistor BC559
2	Germaniumdioden AA119 (D1, D2)
1	Leuchtdiode, rot (LED1, muß vorhanden sein!)
1	Leuchtdiode, Farbe beliebig (LED2)
1	Widerstand RL (z.B. 1 Kiloohm)
1	Widerstand 68 Ohm
1	Widerstand 100 Ohm
1	Widerstand 270 Ohm
1	Widerstand 330 Ohm
3	Widerstände 1 Kiloohm
1	Widerstand 4,7 Kiloohm
1	Widerstand 10 Kiloohm
2	Widerstände 47 Kiloohm
1	Widerstandsarray 8 x 2,2 Kiloohm
15	Kondensatoren 100 Nanofarad, Keramik
2	Kondensatoren 100 Mikrofarad / 16 Volt
1	Nickel-Cadmium-Akku, 2,4...3.6 Volt, lötbar bzw. Trockenbatterie, z.B. Lithium, lötbar
2	Pfostensteckerleisten für Jumper und Steckverbindungen
24	Jumper (Steckbrücken für Pfostenstecker-Pins)
1	lötbare Pfostenbuchsenleiste, 3-pol mit Litze für LED2

Tabelle 5.16: Bauteile für die RAM/ROM-Karte

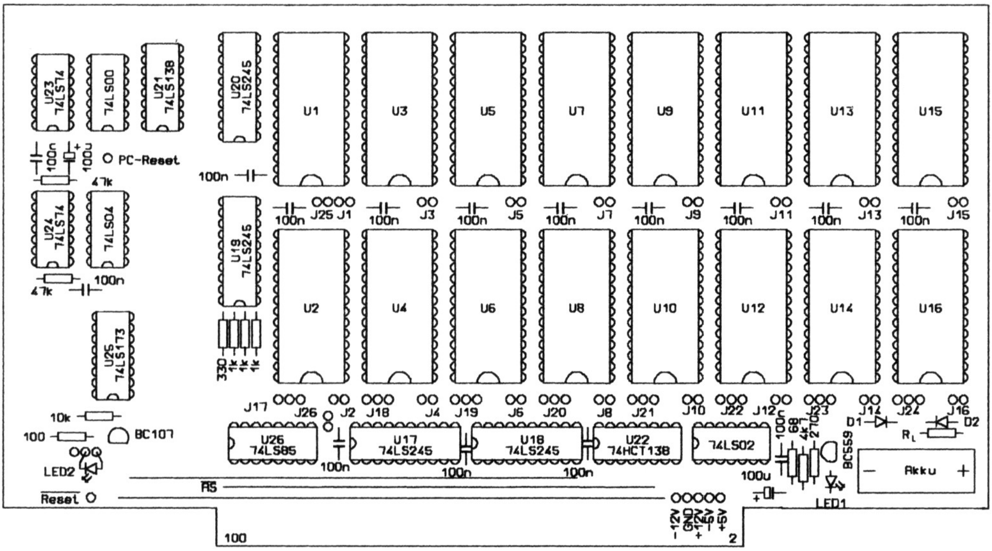


Bild 5.15: Bestückung der RAM/ROM-Karte

(LED1) auszuführen. Diese Leuchtdiode muß im Gegensatz zu LED2 nicht unbedingt angeschlossen werden. Es empfiehlt sich jedoch, sie mit einem Kabel zu versehen, und an gut sichtbarer Stelle im Gehäuse einzubauen. Die dreipolige Verbindung ist symmetrisch belegt, so daß beim Aufstecken nicht auf die Polarität geachtet werden muß.

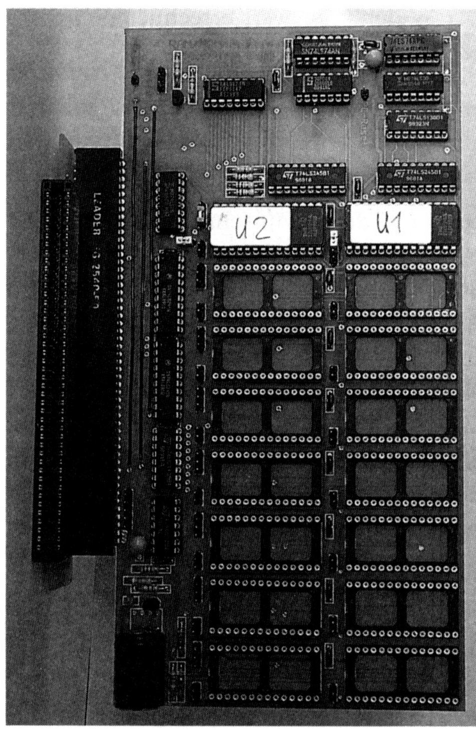


Foto 5.3: RAM/ROM-Karte im Grundausbau

5.3.2.2 Akku und Ladewiderstand

Für den Akku ist eine lötbare Nickel-Cadmium-Miniaturausführung mit mindestens 2,4 Volt zu wählen. Notfalls können auch zwei Zellen mit je 1,2 Volt hintereinandergeschaltet werden. Da zur Erhaltung der Speicherinformationen nur ein sehr kleiner Strom fließt, ist die Kapazität von untergeordnetem Interesse. Stattdessen sollte auf kleine Abmessungen geachtet werden, damit ein problemloser Einbau in die Platine möglich ist.

Während der Einschaltzeit des Rechners wird der Akku über den parallel zu D2 liegenden Widerstand R_L geladen. Der dabei fließende Strom sollte nicht mehr als 1/50 der Akku-Nennkapazität betragen, die seiner Beschriftung zu entnehmen ist. Der Widerstandswert errechnet sich demnach wie folgt:

$$R_L = \frac{4{,}5V - U_a}{I}$$

U_a ist die Spannung des verwendeten Akkus und I der gewünschte Ladestrom. Der ermittelte Wert kann aufgerundet werden. Bei einer Akkuspannung von 2,4 Volt und einer Kapazität von 150mAh ergibt sich I zu (150 / 50)mA = 3mA. R_L ist also (4,5V - 2,4V) / 3mA = 700 Ohm. Verwendet wird in diesem Fall etwa 1 Kiloohm. Ist Ihr Rechner täglich sehr lange in Betrieb, sollte ein größerer Widerstand verwendet werden.

Statt des Akkus kann auch eine Trockenbatterie zum Einsatz kommen. Gut geeignet sind hierzu Lithiumzellen. Sie sind umweltfreundlicher als Nickel-Cadmium-Akkus und enthalten genug Energie für Jahre. Der Ladewiderstand R_L entfällt dann.

D1 und D2 müssen wegen der im Vergleich zu Silizium niedrigeren Durchlaßspannung Germaniumtypen sein. Die Leuchtdiode LED1 muß vorhanden sein und sollte wegen der erforderlichen Durchlaßspannung die Leuchtfarbe Rot besitzen.

5.3.2.3 Bestückung mit Speicherbausteinen und Jumpern

Die Schaltung kann je nach Verwendungswunsch ganz oder auch nur teilweise gemischt mit ROM- und RAM-Bausteinen bestückt und auf mehrere verschiedene Arten betrieben werden. Dabei sind jedoch einige grundlegende Punkte zu beachten:

- Als ROM-Bausteine eignen sich die Typen 27256 mit je 32 Kilobyte Speicherkapazität und 27512 mit je 64 Kilobyte bzw. die entsprechenden CMOS-Typen 27CXXX. Als RAM-Bausteine lassen sich die statischen CMOS-Typen 62256 (entspricht 63256 und 43256) mit je 32 Kilobyte, notfalls auch 6264 mit 8 Kilobyte Fassungsvermögen verwenden. Weil diese ICs eine Datenbusbreite von 8 Bit besitzen, müssen sie immer paarweise eingesetzt werden. Alle Speicherbausteine müssen eine Zugriffszeit von 200 Nanosekunden oder weniger aufweisen.

- In den Sockeln U1 und U2 sollten sich im Normalfall ROMs mit den Autoconfig-Daten befinden. Die Verwendung von RAMs stellt einen Sonderfall dar und wird in Kapitel 5.3.3.5 beschrieben. Die Bestückung von U1 und U2 stellt gleich-

zeitig der Mindestausbau der Karte dar. Die übrigen Speichersockel U3..U16 können leer bleiben oder mit RAMs bzw. ROMs bestückt werden.

- Die Bausteine müssen immer paarweise vom gleichen Typ sein. Das heißt, in untereinanderliegende Sockel (etwa U3, U4 oder U15, U16) dürfen nicht Bausteine mit unterschiedlicher Funktion oder Speichergröße eingesteckt werden. Insbesondere darf nicht ein einzelner Sockel einer Spalte frei bleiben (etwa U3 bestückt, U4 leer).

- Bei gemischter Bestückung mit ROMs und RAMs müssen die ROM-Bausteine von links beginnend, also auf den Sockeln U1, U2; U3, U4; U5, U6... angeordnet werden. RAM-Bausteine dagegen sind von rechts her einzusetzen, also auf den Sockeln U15, U16; U13, U14; U11, U12...

Es darf also kein ROM rechts von einem RAM stecken und kein RAM links von einem ROM.

- Die Jumper müssen je nach Bestückung eingesetzt werden. Es sind insgesamt 24 Brücken zu stecken. Dabei wird zwischen 2-Pin- und 3-Pin-Jumpern unterschieden. Die beiden 2-Pin-Jumper rechts unter den am weitesten rechts befindlichen ROM-Bausteinen bleiben ungesetzt, alle anderen 2-Pin-Jumper sind aufzustecken. Eine Ausnahme bilden die Jumper J25 und J26. Sie sind immer zu stecken, außer beim Einsatz von RAM-Bausteinen in U1, U2. Dieser Sonderfall wird in Kapitel 5.3.3.5 beschrieben. Daneben ist mit den 3-Pin-Jumpern links unter den Sockelspalten für jedes Bausteinpaar die Zuführung der Betriebsspannung vom Netzteil oder vom Akku zu wählen. Steckt der betreffende Jumper links, wird die normale +5V-Versorgung benutzt, steckt er dagegen rechts, erfolgt bei ausgeschaltetem Rechner eine Pufferung über den Akku. Diese Betriebsart ist nur bei CMOS-RAMs sinnvoll.

Ein Beispiel: Sollen vier ROMs und vier RAMs mit dem im Buch enthaltenen Treiber zum Einsatz kommen, dann sind die Sockel U1, U2, U3 und U4 mit den ROMs zu bestücken, U13, U14, U15 und U16 mit den RAMs. Alle 2-Pin-Jumper außer J3 und J4 sind zu stecken. Die 3-Pin-Jumper der CMOS-RAMs J23 und J24 werden rechts gesteckt (Pufferung über den Akku), alle übrigen 3-Pin-Jumper links.

5.3.3 Verwendungsarten der Karte

Der Anschluß der RAM/ROM-Karte an Ihren Amiga wird in Kapitel 5.3.4 beschrieben. Zuvor sollen aber die wichtigsten der vielen möglichen Verwendungsarten und die dazu nötige Bestückung anhand von Beispielen beschrieben werden. Da in den beiden Sockeln U1 und U2 im Normalfall ROM-Bausteine nötig sind, für die Autokonfigurationsdaten aber nur wenig Speicherplatz belegt wird, bietet sich eine Verwendung des freien Bereiches als ROM-Disk an. Ihr Inhalt kann mit einem besonderen Programm zusammengestellt werden, das gleich die Autokonfigurationsdaten und einen Treiber als Grundlage für die vielseitigen Einsatzmöglichkeiten installiert.

5.3.3.1 Betrieb als ROM-Disk

Der einfachste Einsatzfall der Karte ist sicherlich ihr Betrieb als reine ROM-Disk.

Dazu ist sie mit 2 bis 16 ROM-Bausteinen zu bestücken. Die in die ROMs zu brennenden Informationen lassen sich mit dem Programm MakeRomDisk auf einfache Weise aufbereiten. Es wird aufgerufen mit

```
MakeRomDisk Scriptfile [--A]
```

bei Bedarf erst mit Mount angemeldet werden. Weitere Informationen dazu enthält Kapitel 5.3.5.

Die in der ROM-Disk MD0: gewünschten Dateien und ihre Zusammenstellung in Directories sind in einem Scriptfile anzugeben. Hier ein Beispiel:

```
Directory: C

Workbench1.3:C/SetClock   SetClock
C-Compiler:C/cc           cc
C-Compiler:C/as           as
C-Compiler:C/ln           ln

#

Directory: Devs

Workbench1.3:devs/system-configuration   System-Configuration
Workbench1.3:devs/harddisk.device        harddisk.device

Directory: Printers
Workbench1.3:devs/printers/epson   epson
#

Workbench1.3:devs/printer.device   printer.device

#
```

Im aktuellen Directory müssen sich dabei die Dateien MDRom und MemoryDevice befinden. Der optionale Zusatz $\overline{A}$ steht für Autoboot. Bei dieser Angabe wird später das Memory-Device (MD0:, wenn RAM vorhanden auch MD1:) nach dem Einschalten automatisch installiert, so daß auch von dort aus gebootet werden kann. Allerdings besteht diese Möglichkeit nur ab der Betriebssystem-Version Kickstart 1.3 aufwärts. Unter Kickstart 1.2 muß $\overline{A}$ entfallen, da sonst das System abstürzt. Ohne $\overline{A}$ muß das neue Device

Das Scriptfile läßt sich mit jedem Editor erstellen. Dabei sind anstelle von Space auch Tabs verwendbar. Zu jeder Datei ist – ähnlich wie beim Copy-Befehl – deren genaue Herkunft, also der vollständige Pfadname und das Ziel anzugeben. Das Zieldirectory wird einmal mit dem vorangestellten Schlüsselwort »Directory:« benannt. Das File SetClock aus dem Verzeichnis C der Diskette Workbench1.3: und die Files cc, as und ln aus dem Verzeichnis C der Diskette C-Compiler: werden also im Verzeichnis C der ROM-Disk

angelegt. Wie das Beispiel weiter zeigt, sind auch verschachtelte Directories möglich. Das Verzeichnis Devs enthält das Verzeichnis Printers mit dem File epson. Jedes Directory ist mit dem Zeichen # abzuschließen. Aufgrund der Syntax darf keine Quelldiskette mit Namen Directory: verwendet werden!

Das Programm MakeRomDisk stellt aufgrund der Beschreibung im Scriptfile die Informationen für die EPROMs zusammen. Dabei fordert es gegebenenfalls dazu auf, die benannten Disketten einzulegen.

Es erzeugt für jedes EPROM ein File, das anschließend mit dem in Kapitel 2.8 vorgestellten (oder mit jedem anderen) EPROM-Programmierer in den vorgesehenen Baustein gebrannt werden muß. Dabei werden ausschließlich die EPROM-Typen 27512 unterstützt. Die Zuordnung der Bausteine zu den Sockeln U1..U16 geht aus den entsprechenden Filenamen hervor. Die Bausteine sind auf den angegebenen Plätzen zu bestücken und die Jumper nach Anweisung des Programms zu stecken. Grundsätzlich sind die beiden 2-Pin-Jumper rechts unter der letzten EPROM-Reihe offenzulassen. Die 3-Pin-Jumper aller mit EPROM-Bausteinen bestückten Sockel müssen sich auf der linken Position (keine Akkupufferung) befinden. Zusätzlich werden bei Betrieb ohne Autoboot noch die Daten für die Mountlist ausgegeben. Weitere Informationen dazu enthält Kapitel 5.3.5.

5.3.3.2 Betrieb als RAM- und ROM-Disk

Sind nicht alle Sockel mit ROMs belegt, lassen sich die übrigen ganz oder auch nur zum Teil paarweise mit RAM-Bausteinen bestücken. Der beim Zusammenstellen des ROM-Inhalts mit MakeRomDisk bereits installierte RAM/ROM-Disk-Treiber erkennt automatisch, wie viele RAMs eingesetzt wurden. Allerdings unterstützt er dabei nur Bausteine vom Typ 62256. Weil mindestens für U1 und U2 ROMs vorzusehen sind, ist die maximal nutzbare RAM-Disk-Größe bei Verwendung der Sockel U3..U16 14 mal 32 Kilobyte, also 448 Kilobyte.

Der Inhalt der EPROMs wird auch hier mit dem zuvor beschriebenen Programm MakeRomDisk generiert.

Alle ROMs müssen durchgehend links, also auf den Sockeln mit den niedrigeren Nummern stecken, wie vom Generierprogramm gefordert. Die RAMs dagegen müssen von rechts her angeordnet werden. Dabei darf zwischen zwei RAM-Bänken kein Sockelpaar ausgelassen werden, weil der Treiber ansonsten die folgenden Bausteine nicht mehr für die RAM-Disk verwendet.

Die 2-Pin-Jumper J15 und J16 sind immer beim Einsatz von RAMs zu stecken, die 2-Pin-Jumper rechts unter den beiden letzten ROM-Bausteinen dagegen müssen herausgenommen werden.

Sollen die Informationen in den CMOS-RAMs auch nach dem Ausschalten des Rechners noch vorhanden bleiben, sind die 3-Pin-Jumper unter den entsprechenden RAM-Paaren in die rechte Position zu bringen. Nähere Informationen über Vorbereitung und Einsatz der RAM-Disk enthält Kapitel 5.3.5.

5.3.3.3 Gleichzeitiger Betrieb als ROM-Disk, RAM-Disk und Speichererweiterung

Der RAM/ROM-Disk-Treiber überprüft bei seiner Installation, ob und wieviele RAM-Bausteine sich lückenlos von rechts her in den Sockeln U16, U15, U14, U13... befinden, um die Größe der RAM-Disk bestimmen zu können. Beim ersten leeren oder mit einem ROM bzw. mit einem PC-Adapter aus Kapitel 5.3.3.6 bestückten Sockelpaar bricht er den Testvorgang ab.

Soll ein Teil des RAM-Vorrats nicht für die RAM-Disk verwendet werden, ist einfach bei der Bestückung zwischen den ICs für die RAM-Disk und denen für die Speichererweiterung ein Sockelpaar freizulassen. Die nicht vom Treiber erkannten RAMs lassen sich als zusätzlicher System-RAM-Bereich verwenden.

Soll also eine ROM-Disk mit 512 Kilobyte und eine reset- und ausschaltfeste RAM-Disk mit 128 Kilobyte und zusätzlich eine System-Speichererweiterung von 192 Kilobyte installiert werden, sind die Sockel U1..U4 mit EPROMs vom Typ 27512 zu bestücken, in U16..U13 kommen vier RAMs vom Typ 62256, die vom Treiber für die RAM-Disk verwendet werden, U11 und U12 bleiben frei und in U5.. U10 werden nocheinmal 6 RAMs vom Typ 62256 gesteckt. Die Lage der 2-Pin-Jumper ist hier wieder J3 und J4 offen, alle anderen gesteckt. Bei den 3-Pin-Jumpern für die Betriebsspannung sind J17 und J18 links zu stecken (normale +5V-Versorgung), alle anderen können sich rechts befinden, um Akku-Pufferung der jeweiligen RAMs zu erreichen.

Nach dem Hochfahren des Rechners wie gewöhnlich muß der zusätzliche RAM-Vorrat dem System noch mitgeteilt werden. Dies geschieht mit dem im Verzeichnis C der Diskette zum Buch enthaltenen Programm Addmem, das als Parameter Anfangs- und Endadresse des einzubindenden Speicherbereiches erwartet. Die Anfangsadresse jeder Speicherbank ist im Schaltplan (Bild 5.14) vermerkt, die Endadresse ist hier die erste nicht mehr zum Bereich gehörige Adresse. Soll das mit zwei 32-Kilobyte-Bausteinen 62256 bestückte Sockelpaar U5, U6 (Basisadresse $X40000) dem System bekannt gemacht werden, muß der Befehl

```
Addmem X40000 X50000
```

lauten, denn die Speichergröße beträgt 2 mal 32, also 64 Kilobyte. Das entspricht einer Länge von $10000. X steht für die bei der Konfiguration festgelegte Basisadresse der RAM/ROM-Karte. Beim Einstecken als erste Karte ist sie normalerweise $200000 und damit X also 2. Die Basisadresse der RAM/ROM-Karte kann durch Aufruf des Programms RAMROM-base ermittelt werden, wenn zur Erstellung der EPROM-Daten das Programm MakeRomDisk verwendet wurde.

Da jeder Sockel zur Aufnahme von 64-Kilobyte-Bausteinen vorgesehen ist, entsteht beim Einsatz von mehreren 32-Kilobyte-RAMs kein durchgehender Speicherbereich. Allerdings erscheint jeder 32-Kilobyte-Baustein zweimal nacheinander im Adreßraum, weil die höchstwertige Adreßleitung von ihm ja nicht ausgewertet wird. Diese Eigenschaft der Karte kann man sich zunutze machen, in-

dem man für je zwei aufeinanderfolgende Speicherbänke als Beginn des RAM-Bereiches nicht die Basisadresse des ersten Sockelpaares, sondern den Beginn des zweiten Bereiches dort angibt. So entsteht ein durchgehender Speicherblock bis zum Ende des ersten Bereiches in der nachfolgenden Speicherbank. Die Länge ist dann 4 mal 32, also 128 Kilobyte, oder hexadezimal $20000. Für U7, U8, U9, U10 bestückt mit RAMs vom Typ 62256 lautet also der entsprechende Befehl

```
Addmem X70000 X90000
```

Statt der RAMs 62256 ist für die Speichererweiterung zur Not auch die 8-Kilobyte-Ausführung 6264 verwendbar. Bei ihr werden A14 und A15 nicht ausgewertet und A13 muß HIGH sein. Daraus ergibt sich, daß ein solches Bausteinpaar viermal im Adreßraum des Sockels auftaucht, und zwar in den Bereichen $4000..$7FFF, $C000..$FFFF, $14000..$17FFF und $1C000..$1FFFF relativ zur Basisadresse. Die Einbindung von zwei 6264-Bausteinen in den Sockeln U5 und U6 kann also mit

```
Addmem X64000 X68000
```

erfolgen. Da sich hier kein durchgehender Adreßbereich ergibt, können nicht mehrere Sockelpaare zusammengefaßt werden.

5.3.3.4 Betrieb als Speichererweiterung ohne RAM-Disk

Auch bei dieser Betriebsart müssen sich in den Sockeln U1 und U2 wie immer die Konfigurations-Daten befinden. Sie werden durch das Programm MakeRomDisk in den ROM-Inhalt für U1 und U2 eingebunden. Es empfiehlt sich also, gleich einige Files zusammenzustellen und sie in zwei ROMs unterzubringen. Soll kein Baustein für die RAM-Disk erkannt werden, sind die beiden Sockel U15 und U16 einfach freizulassen. Damit sind mit 12 RAM-Bausteinen Erweiterungen von maximal 6 mal 64, also 384 Kilobyte möglich. Die Einbindung des zusätzlichen Speichers in das System geschieht wieder mit dem Befehl Addmem, wie in Abschnitt 5.3.3.3 beschrieben.

Die Möglichkeit, alle freien Sockel mit RAM für eine Speichererweiterung zu bestücken, ergibt sich generell unter der Betriebssystemversion Kickstart 1.2, oder wenn unter Kickstart 1.3 auf automatisches Installieren verzichtet wird ($\overline{A}$ beim Aufruf von MakeRomDisk fortlassen). Damit ist ein wahlweiser Betrieb möglich: Entweder wird die RAM-Disk mit Mount DM1: angemeldet, oder die RAMs mit dem zuvor beschriebenen Befehl Addmem als Systemspeichererweiterung eingebunden. Allerdings darf nie beides zugleich geschehen. Die maximale RAM-Größe ist dann mit 14 Bausteinen 7 mal 64, also 448 Kilobyte.

5.3.3.5 Einsatz von RAMs in U1 und U2

Normalerweise sind in den Sockeln U1 und U2 ROMs vorzusehen, in denen sich die Konfigurationsdaten und nach dem Konfigurieren selbsttätig auszuführende Programme befinden, wie etwa der RAM/ROM-Disk-Treiber. Gerade für die Softwareentwicklung ist es jedoch wünschenswert, auch hier RAMs zu installieren, um leicht Testläufe durchführen zu

können. In diesem Fall müssen J25 und J26 offen bleiben und alle anderen 2-Pin-Jumper gesteckt werden. Es kann kein zusätzliches ROM zum Einsatz kommen.

Für einen störungsfreien Betrieb müssen beim Einschalten unbedingt sinnvolle Autokonfigurationsdaten vorhanden sein. Sind die entsprechenden Angaben offensichtlich falsch, was in der Regel bei dem zufälligen Inhalt noch unbeschriebener RAMs der Fall ist, wird das Board vom System gar nicht erst konfiguriert, das heißt, es wird keine Basisadresse in das Register U25 geschrieben. Damit wird auch das Config-Flipflop nicht gesetzt und der Anschluß $\overline{\text{CONFIGOUT}}$ bleibt HIGH, was dazu führt, daß weitere Karten in der Konfigurationskette nicht erkannt werden. Trotzdem bleibt das Basisadreßregister unter der Adresse $E80048 ansprechbar. Eine neue Basisadresse für die Karte kann also von Hand eingestellt werden. Das System legt die erste Zusatzkarte der Konfigurationskette im Normalfall nach $200000. Von Hand geschieht das durch Schreiben von $20 auf die Adresse $E80048.

Beim Einsatz von akkugepufferten CMOS-RAMs in U1 und U2 sind nach dem Einschalten die zuletzt eingeschriebenen Daten noch vorhanden. Enthalten diese aber einen Fehler, kann das zur Folge haben, daß sich das System nicht hochfahren läßt. In einem solchen Fall muß durch Abziehen von J17 die Stromversorgung der Speicherbausteine für einige Zeit unterbrochen werden.

Damit beim Beschreiben der RAMs nicht der Zustand der ebenfalls dort lokalisierten Register verändert wird, dürfen Schreibzugriffe auf den gesamten Bereich der Sockel U1 und U2 nur wortweise erfolgen. Das auf der beiliegenden Diskette im Verzeichnis C enthaltene Programm WriteFile von Bernhard Möllemann schreibt Daten aus einem File in den angegebenen RAM-Bereich. Standardmäßig geschieht das wortorientiert. Das Befehlsformat ist

```
WriteFile Filename -hex-Adresse
```

Die unter Filename abgelegten Daten werden ab der angegebenen Adresse wortweise in den Speicher geschrieben. Eine Auflistung weiterer möglicher Optionen erhält man mit WriteFile ?.

5.3.3.6 Ansteuerung von PC-I/O-Zusätzen

Für keinen Rechner existieren so viele verschiedene Zusatzkarten wie für den IBM-PC und seine vielen Stiefbrüder. In den meisten Fällen handelt es sich dabei um I/O-Schaltungen wie Meßdatenerfassung und Interfaceschaltungen. Aufgrund der hohen Stückzahlen sind diese Zusätze abgesehen von der Vielfalt des Angebots meist vergleichsweise preisgünstig erhältlich. Wie sich diese Zusätze auch ohne PC-Brückenkarte am Amiga einsetzen lassen, soll nun beschrieben werden.

Kabel reicht schon!

An die Sockel können allein über ein Adapterkabel direkt für den IBM-PC konzipierte I/O-Schaltungen angeschlossen werden. Die dabei nötigen Verbindungen zeigt Tabelle 5.17.

IBM-Slot (z.B. für OMTI)		RAM/ROM-Sockel	
Signalname	Pin	Pin	Signalname
$\overline{\text{I/O CHCK}}$	A1		
D7	A2	19	D7
D6	A3	18	D6
D5	A4	17	D5
D4	A5	16	D4
D3	A6	15	D3
D2	A7	13	D2
D1	A8	12	D1
D0	A9	11	D0
I/O CHRDY	A10		
AEN	A11	20	$\overline{\text{CS}}$
A19	A12	14	GND
A18	A13	14	GND
A17	A14	14	GND
A16	A15	14	GND
A15	A16	14	GND
A14	A17	14	GND
A13	A18	14	GND
A12	A19	14	GND
A11	A20	14	GND
A10	A21	14	GND
A9	A22	24	A9
A8	A23	25	A8
A7	A24	3	A7
A6	A25	4	A6
A5	A26	5	A5
A4	A27	6	A4
A3	A28	7	A3
A2	A29	8	A2
A1	A30	9	A1
A0	A31	10	A0

IBM-Slot (z.B. für OMTI)		RAM/ROM-Sockel	
Signalname	Pin	Pin	Signalname
GND	B1	14	GND
Reset	B2	Reset-Pin	
+5V	B3		
IRQ2	B4	14	GND
–5V	B5		
DRQ2	B6	14	GND
–12V	B7		
CRD SLCTD	B8		
+12V	B9		
GND	B10		
$\overline{\text{MEMW}}$	B11		
$\overline{\text{MEMR}}$	B12		
$\overline{\text{IOW}}$	B13	27	$\overline{\text{WE}}$
$\overline{\text{IOR}}$	B14	22	$\overline{\text{OE}}$
$\overline{\text{DACK3}}$	B15		
DRQ3	B16		
$\overline{\text{DACK1}}$	B17		
DRQ1	B18		
$\overline{\text{DACK0}}$	B19		
CLOCK	B20		
IRQ7	B21		
IRQ6	B22		
IRQ5	B23		
IRQ4	B24		
IRQ3	B25		
$\overline{\text{DACK2}}$	B26		
T/C	B27		
ALE	B28		
+5V	B29		
OSC	B30		
GND	B31		

Tabelle 5.17: Belegung des Adapterkabels zum Anschluß von PC-Platinen an die RAM/ROM-Karte

Das von der Schaltung generierte Signal $\overline{\text{WE}}$ entspricht dem PC-Signal $\overline{\text{IOW}}$. Ebenso entspricht $\overline{\text{OE}}$ der Information $\overline{\text{IOR}}$. $\overline{\text{CS}}$ wurde mit der Leitung $\overline{\text{AEN}}$ des PC-Bus verbunden. Das wichtige Signal Reset liegt nicht an den RAM/ROM-Sockeln. Es ist im Gegensatz zur Reset-Leitung des Amiga HIGH-aktiv. Daher wird es auf der RAM/ROM-Karte an RESETSEL abgegriffen, was den Vorteil

hat, daß ein Reset aller Zusätze am PC-Bus außer bei einem Systemreset auch durch einen Bytezugriff auf Adresse \$X00030 erreicht werden kann. (X steht für die bei der Konfiguration festgelegte Basisadresse.) Das dekonfiguriert zwar die RAM/ROM-Karte und sperrt durch $\overline{\text{CONFIGOUT}}$ = HIGH alle nachfolgenden Karten in der Konfigurationskette, doch das Schreiben der ursprünglichen Adresse X in das Basisadreßregister unter \$E80048 macht diesen Effekt schnell wieder rückgängig. Das so gewonnene Reset-Signal für den PC-Bus wird an dem Steckkontakt PC-Reset links oben auf der Karte zur Verfügung gestellt. Zur Stromversorgung der PC-Zusätze über die RAM/ROM-Karte wurden die am Slot vorhandenen Spannungen +5 Volt, –5 Volt, +12 Volt und –12 Volt sowie Masse rechts über den Steckkontakten auf eine Verbindungsleiste herausgeführt.

Als Verbindung zwischen dem Sockel und der PC-I/O-Karte sollte ein Flachbandkabel eingesetzt werden. Bei langen Ausführungen empfiehlt es sich, jede zweite Ader an Masse zu legen.

Ein komfortabler PC-Bus-Adapter

Statt des Flachbandkabels, das mit seinen vielen Verbindungen nicht einfach zu konfektionieren ist, kann auch die im folgenden vorgestellte PC-Bus-Adapterplatine Verwendung finden. Sie wird ebenfalls an einen RAM-Steckplatz der RAM/ROM-Karte angeschlossen und bietet zusätzlich den Vorteil, daß nicht der gesamte Adreßraum des verwendeten Sockels für PC-Zwecke verlorengeht. Der RAM-Baustein aus dem vom Adapterkabel belegten Sockel kann auf der Adapterplatine eingesteckt werden. Lediglich ein Kilobyte wird daraus zur Emulierung des PC-I/O-Adreßraums ausgeblendet.

Die Dekodierung der Bereiche übernimmt die auf der Adapterplatine enthaltene Logik. Beachten Sie dazu den Schaltplan in Bild 5.16. Beim hier vorgesehenen Einsatzfall verbindet der Jumper J2 immer den Anschluß $\overline{\text{OE}}$ mit dem Eingang des OR-Gatters. Der Ausgang des achtfach-NAND-Gatters 74LS30 wird nur dann LOW, wenn eine Adresse aus dem für den PC-Bus vorgesehenen oberen 2-Kilobyte-Bereich (CPUA11..CPUA15 = HIGH) am RAM-Sockel anliegt. Ist das nicht der Fall, wird über den nachgeschalteten Negierer das OR-Gatter freigegeben und das Signal $\overline{\text{CS}}$ beim Eintreffen vom Sockel auf der RAM/ROM-Karte direkt zum Anschluß $\overline{\text{CS}}$ am RAM-Sockel der Adapterplatine durchgeschaltet. Liegt jedoch eine Adresse mit CPUA11..CPUA15 = HIGH auf dem Bus, gelangt das $\overline{\text{CS}}$-Signal über das andere OR-Gatter zur PC-Logik. Dort wird zunächst über Negierer und Transistor eine Signal-LED angesteuert, die aufleuchtet, sobald der PC-Bus selektiert ist. Gleichzeitig wird der Zustand des Steuersignals $\overline{\text{WE}}$ auf den PC-Bus-Anschluß $\overline{\text{IOW}}$ und $\overline{\text{OE}}$ auf $\overline{\text{IOR}}$ durchgeschaltet.

Die Platine (Bild 5.17) ist so ausgeführt, daß sie direkt in einen PC-Slot des Amiga 2000 paßt. Es bleiben drei PC-Slots zum Einstecken von PC-I/O frei. So kann die vorhandene Bus-Hardware auch ohne eine PC- oder gar AT-Brückenkarte genutzt werden. Natürlich ist dann der Betrieb entweder mit dem vorgestellten Adapter oder mit einer Brückenkarte möglich.

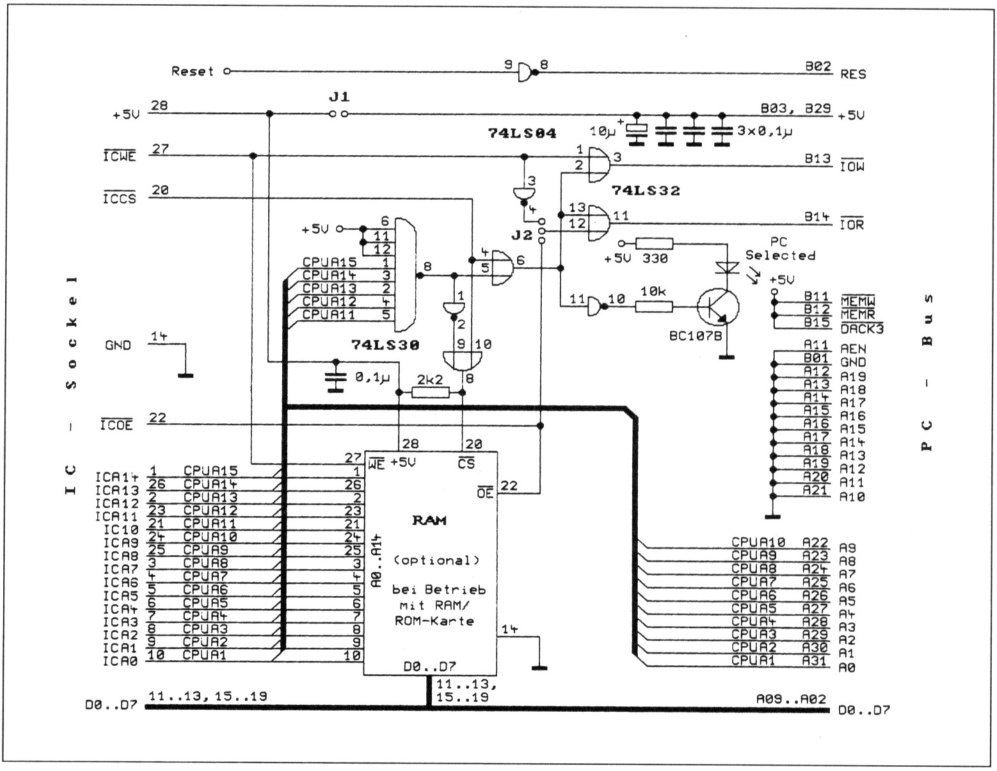


Bild 5.16: Schaltplan des PC-Adapters

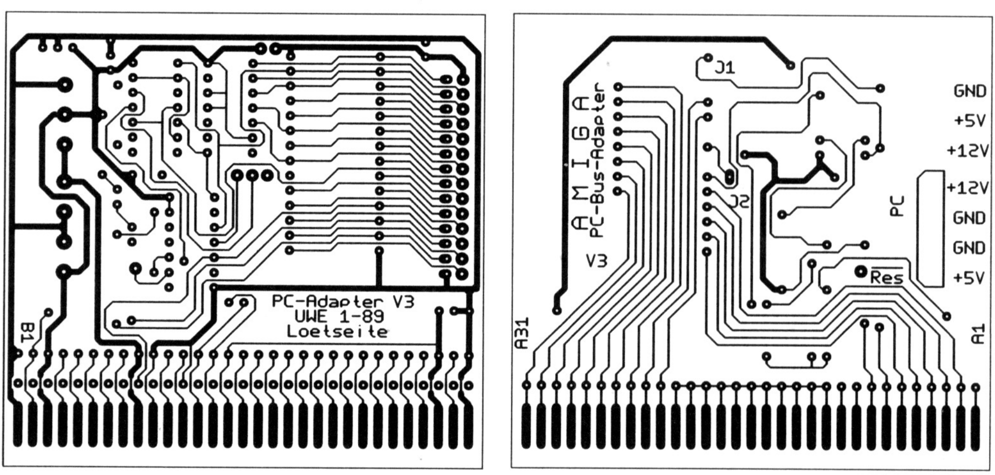


Bild 5.17: Layout für den Bus-Adapter

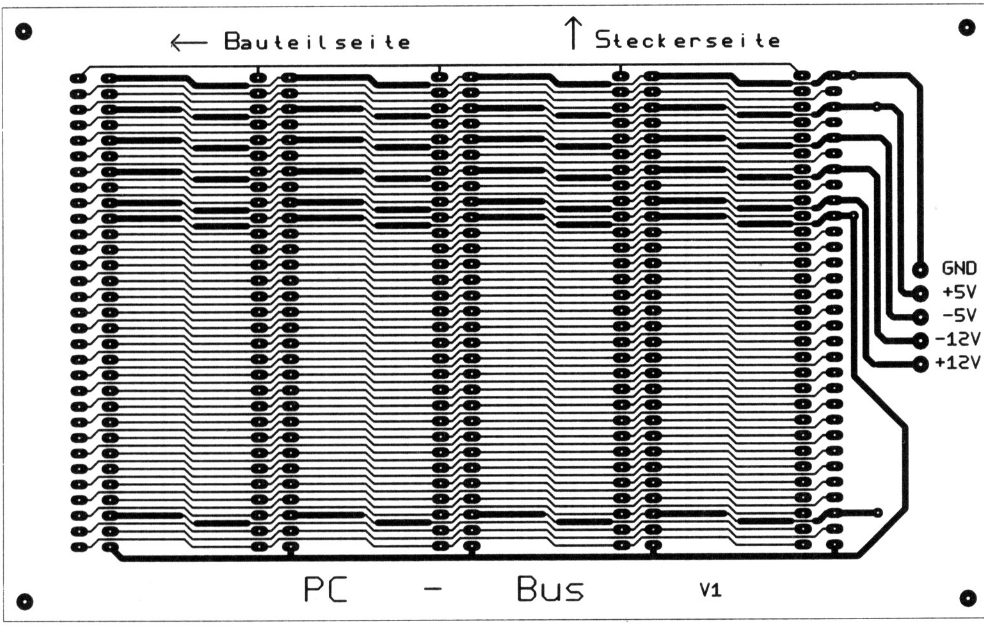


Bild 5.18: Layout für die PC-Bus-Platine

Beim Amiga 500 und 1000 sowie beim Amiga 2000 mit eingesteckter Brückenkarte läßt sich direkt auf der Adapterplatine ein PC-Slot unterbringen, in den die gewünschte PC-Zusatzkarte eingesteckt wird. Sollen mehrere Platinen betrieben werden, muß die PC-Bus-Karte nach Bild 5.18 und dem Bestückungsplan in Bild 5.19 aufgebaut werden. Sie entspricht dem PC-Bus beim Amiga 2000 und bietet Platz für insgesamt fünf PC-Slots. In einen davon wird die PC-Adapterplatine gesteckt. Genau wie beim Amiga 2000 bereitet dann der Betrieb mehrerer PC-I/O-Karten, zum Beispiel eines OMTI-Harddisk-Controllers und einer Transputer-Link-Interfaceplatine, vom Amiga-Bus aus mit entsprechender Software keine Schwierigkeiten.

Die doppelseitige PC-Adapterplatine nach dem Layout in Bild 5.17 wird wie üblich nach dem Bestückungsplan in Bild 5.20 und den Hinweisen in Anhang A bestückt. Die nötigen Bauteile führt Tabelle 5.18 auf. Um auch die nötigen Lötverbindungen von der Bestückungsseite her ausführen zu können, sind geeignete Sockel, zum Beispiel solche mit gedrehten Beinchen, zu verwenden und es ist eine bestimmte Reihenfolge einzuhalten. Beginnen Sie mit den sechs Durchkontaktierungen, indem Sie Drähtchen durch die entsprechenden Lötaugen stecken und sie von oben und unten verlöten. Danach folgt der Einbau des PC-Slots. Er ist nicht ganz bis zum Anschlag einzustecken, so daß an der Platinenoberseite genügend Platz für die Lötspitze bleibt, denn die

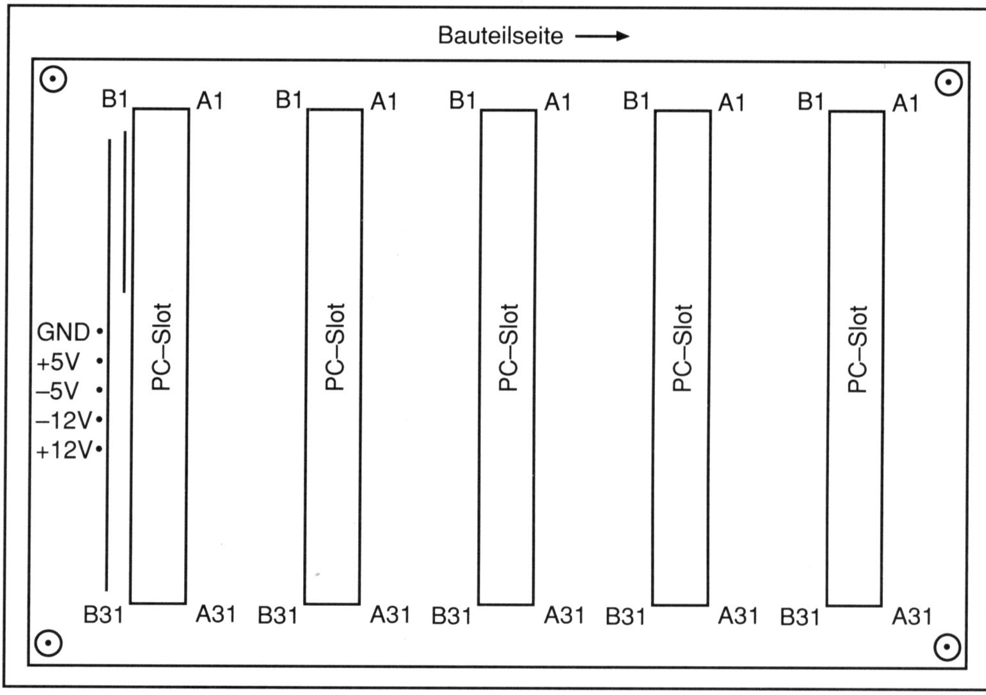


Bild 5.19: Bestückung der PC-Slot-Platine

Anschlüsse der äußeren Reihe (A1..A31) sowie drei Pins der gegenüberliegenden Seite (B11, B12 und B15) sind auch von oben her zu verlöten. Wird der Slot nicht bestückt, dürfen diese Durchkontaktierungen nicht vergessen werden. Danach folgt der 28-polige RAM-Sockel. Er ist wegen der Durchkontaktierungen zu bestücken, auch wenn der RAM-Baustein selbst zunächst nicht eingesetzt werden soll. Jetzt folgt der rechte 14-polige Sockel für die OR-Gatter 74LS32, danach der linke für das NAND-Gatter 74LS30 und schließlich der untere für die Negierer 74LS04. Bei jedem Baustein müssen die vorgesehenen Kontakte von der Bestückungsseite her gelötet werden, sonst besteht die Gefahr, daß sie später nicht mehr zugänglich sind. Kontrollieren Sie nach jedem Schritt mit einem Meßgerät auf Kurzschlüsse.

Der Transistor und die Signal-LED können entfallen, wenn auf eine Selektierungsanzeige keinen Wert gelegt wird. Sie sind jedoch beim Testen der Schaltung und der Software sehr nützlich. Der Pull-up-Widerstand vom 2,2 Kiloohm wird nur in Spezialfällen nötig. Er bleibt normalerweise unbestückt. Beim Flachbandkabel zum Anschluß an einen Sockel auf der RAM/ROM-Karte wurden auf beiden Seiten Stecker mit Quetschverbindungen vorgesehen, um den Lötaufwand zu begrenzen. Die 28-polige Pfostenbuchsen-

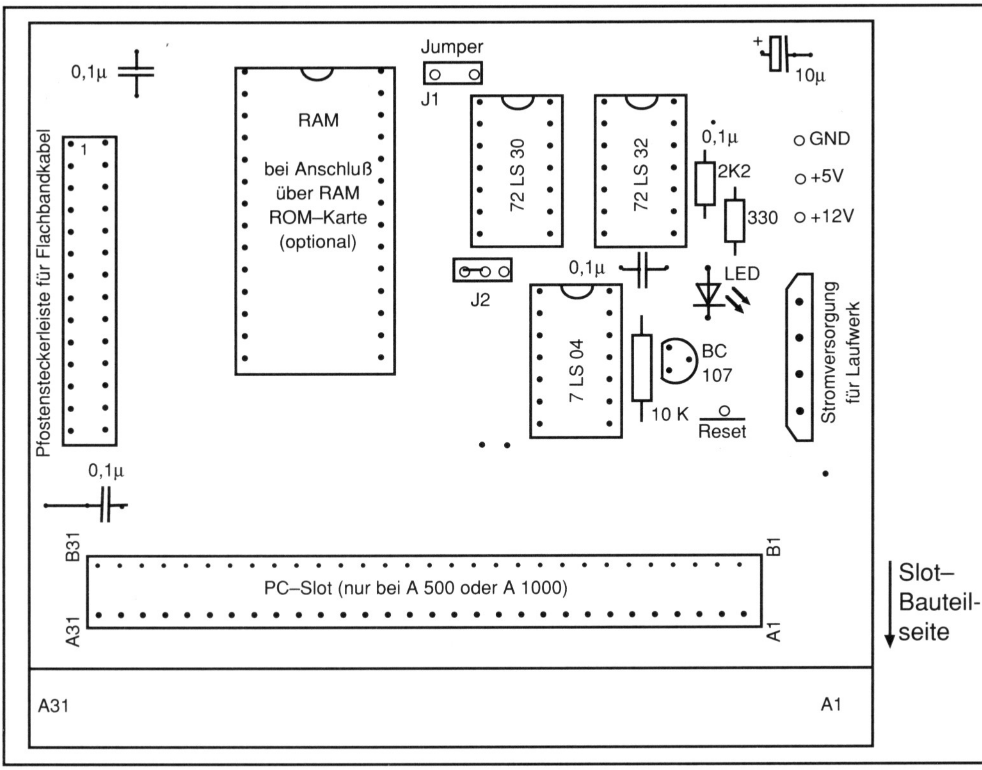


Bild 5.20: Bestückung der Platine für den PC-Slot

leiste ist leider nur schwer zu bekommen. Es lassen sich jedoch auch größere Ausführungen verwenden. Beachten Sie beim Einstecken des Flachbandkabels in einen Sockel der RAM/ROM-Karte unbedingt die richtige Lage. Die Daten- und Adreßleitungen des IC-Sockels auf der PC-Adapterplatine müssen direkt mit denen auf der RAM/ROM-Karte verbunden sein. Messen Sie das vorsichtshalber nach!

Das Reset-Signal wird – anders als bei der reinen Kabel-Lösung – hier am Punkt $\overline{\text{Reset}}$ unten links auf der RAM/ROM-Karte abgegriffen und mit dem Punkt $\overline{\text{Res}}$ auf der Adapterplatine verbunden.

Beim hier beschriebenen Einsatz mit der RAM/ROM-Karte ist der 3-Pin-Jumper J2 auf der Adapterplatine immer links zu stecken, so daß $\overline{\text{IOR}}$ aus $\overline{\text{OE}}$ generiert wird.

Die Stromversorgung des RAM-Sockels auf der Adapterplatine erfolgt in jedem Fall von der RAM/ROM-Karte her. So ist auch Akkupufferung möglich. Soll der Adapter ebenfalls vom Sockel her mit +5 Volt versorgt werden, ist der vorgesehene Jumper J1 zu schließen. In diesem Sonderfall darf keine Akkupufferung erfolgen. Normalerweise bleibt J1 offen, insbesondere wenn +5 Volt von der ange-

1	doppelseitige Platine nach Bild 5.17
1	6-fach NOT 74LS04
1	NAND 74LS30
1	4-fach OR 74LS32
1	IC-Sockel 28-pol
3	IC-Sockel 14-pol
1	Transistor BC107B
1	Leuchtdiode, Farbe nach Wahl
1	Widerstand 330 Ohm
1	Widerstand 2,2 Kiloohm (entfällt normalerweise)
1	Widerstand 10 Kiloohm
4	Kondensatoren 100 Nanofarad, Keramik
1	Elektrolytkondensator 10 Mikrofarad / 16 Volt
1	Pfostensteckerleiste min. 28-pol, zweireihig
1	Pfostensteckerleiste min. 5-pol, einreihig für Jumper
2	Jumper (Kurzschlußstecker für Pfostenkontakte)
1	PC-Slot, 2 x 31-pol
1	Pfostenbuchsenleiste, min. 28-pol mit Quetschkontakten für Flachbandkabel
1	DIL-Stecker, 28-pol mit Quetschkontakten für Flachbandkabel, min. 28-pol Schaltlitze
7	Lötnägel

Tabelle 5.18: Die Bauteile für den PC-Bus-Adapter

schlossenen Schaltung zugeführt wird. Das ist beim Amiga 2000 der Fall, weil die Betriebsspannungen bereits über die Hauptplatine anliegen. Beim Amiga 500 und 1000 ist eine Versorgung über ein externes Netzteil – beispielsweise gemeinsam mit einer eventuell angeschlossenen Harddisk – angebracht. Dazu wurden auf der rechten Seite der Adapterplatine entsprechende Steckverbinder vorgesehen. Oben kann mit +5 Volt, +12 Volt und Masse die Verbindung zum Netzteil hergestellt werden, unten ist Platz zum Einbau eines Normsteckers für Printmontage zur Versorgung eines Hard-Disk-Laufwerkes. Bei der Bus-Platine nach Bild 5.18 erfolgt die Stromversorgung über die dort vorgesehenen Steckkontakte. In allen diesen Fällen bleibt J1 offen.

Die sich beim Einsatz des PC-Bus-Adapters ergebende Aufteilung des Sockel-Adreßbereichs bezüglich der Konfigurations- und Sockel-Basisadresse (siehe Kapitel 5.3.3.6) zeigt Tabelle 5.19.

Die Basisadresse der RAM/ROM-Karte läßt sich durch Aufrufen des Programms RAMROMbase ermitteln. Weil der für den Anschluß des PC-Bus-Adapters verwendete Sockel der RAM/ROM-Karte als RAM-Platz gejumpert sein muß, kann die höchstwertige Adreßleitung nicht ausgewertet werden. Dadurch erscheinen die Bereiche doppelt. Für die PC-Bus-

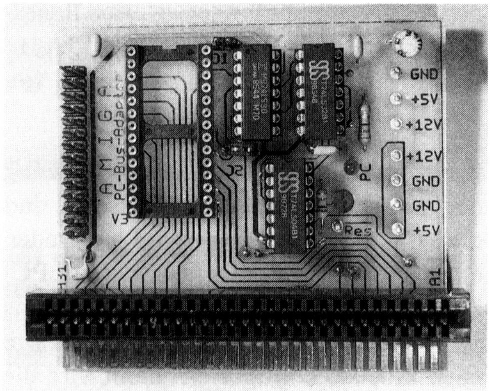


Foto 5.4: PC-Adapter mit einem Slot

Adressen	Nutzung
$XX0000 . . . $XX77FF	RAM
$XX7800 . . . $XX7FFF	PC-Bus-Emulation
$XX8000 . . . $XXF7FF	RAM
$XXF800 . . . $XXFFFF	PC-Bus-Emulation

Tabelle 5.19: Lage des RAM- und des PC-Bus-Bereiches beim PC-Bus-Adapter

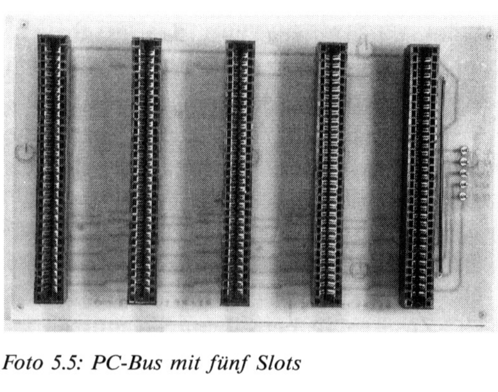


Foto 5.5: PC-Bus mit fünf Slots

Emulation wird aus dem RAM-Bereich des Adapters ein Kilobyte ausgeblendet. Dadurch ist der korrespondierende Speicherbereich im zugehörigen zweiten Sockel auf der RAM/ROM-Karte für normale Anwendungen ebenfalls nicht nutzbar. Wird also der RAM-Sockel auf der Adapterplatine und der parallelliegende Sockel auf der RAM/ROM-Karte jeweils mit einem Baustein vom Typ 62256 bestückt, lassen sich 62 der vorhandenen 64 Kilobyte für Speichererweiterungen benutzen. Das Einbinden erfolgt wieder mit

einem Addmem-Befehl, also beispielsweise mit

```
Addmem 260000 26780
```

für die nach $200000 konfigurierte RAM/ROM-Karte mit PC-Bus-Adapter, mit einem RAM-Baustein 62256 in Sockel U7 und einem weiteren RAM-Baustein 62256 im Sockel U8.

PC-Bus-Emulation am Beispiel eines Hard-Disk-Controllers

Das Adapterkabel – gleichgültig ob direkt verbunden (Seite 234) oder zur Adapterplatine nach Seite 237 – ist in einen freien Sockel der oberen Reihe (niederwertige Datenbushälfte!) zu stecken, der als RAM-Steckplatz gejumpert werden muß. Je nach verwendetem Sockel ergibt sich eine eigene Basisadresse für die externen Zusätze. Bei der reinen Kabellösung nach Seite 237 taucht der PC-Adreßraum mehrfach auf. Aus Gründen der Kompatibilität sollte aber auch dabei der Adapter-Bereich ab $XXF800 angesprochen werden. Tabelle 5.20 enthält die sich ergebenden PC-Basisadressen bezüglich der bei

Sockel	PC-Bus-Basisadresse
U1 / U2	$X0F800
U3 / U4	$X2F800
U5 / U6	$X4F800
U7 / U8	$X6F800
U9 / U10	$X8F800
U11/ U12	$XAF800
U13/ U14	$XCF800
U15/ U16	$XEF800

Tabelle 5.20: Basisadressen der acht RAM/ROM-Sockelpaare

der Autokonfiguration festgelegten Board-Adresse (X). Die Basisadresse der RAM/ROM-Karte kann durch Aufrufen von RAM/ROMbase ermittelt werden.

Aufgrund der ausschließlichen Nutzung der niederwertigen Datenbushälfte sind die PC-Zusätze hier immer auf ungeraden 68000-Adressen ansprechbar. Ist ein PC-Zusatz beispielsweise an Sockel U7 (Basis $X6) angeschlossen und wurde die Karte nach $200000 konfiguriert, dann wird die I/O-Adresse n des PC-Bereichs durch Ansprechen der 68000-Adresse $26F801 + (2 * n) erreicht. Die PC-I/O-Adresse $000 liegt also bei $26F801 im Amiga-Adreßraum, die I/O-Adresse $001 bei $26F803 und so fort. Der Basisadresse des Hard-Disk-Controllers $320 entspricht dann die Amiga-Adresse ($26F801 + 2 * $320), also $26FE41.

Inzwischen gibt es bereits mehrere Treiber für den Amiga, die einen OMTI 5520 oder 5527 Festplatten-Controller bzw. eine kompatible Seagate-Lösung unterstützen. Beispiele dafür sind A.L.F. oder die in c't 3/88 und c't 2/89 vom Heise-Verlag vorgestellte Lösung. In den entsprechenden Treibern müssen lediglich die Adressen der OMTI-Register geändert werden. Beim c't-Treiber kann das geschehen, indem mit einem File-Editor nach dem Auftauchen der Adresse $810640 bzw. $EF0640 gesucht und diese durch die Basisadresse des verwendeten Sockels plus $FE40 ersetzt wird.

Natürlich ist es auch möglich, neben einem Festplatten-Controller noch andere PC-Zusätze zu betreiben. Sie müssen jedoch im PC-I/O-Adreßraum liegen, also

über die PC-Signale $\overline{\text{IOR}}$ und $\overline{\text{IOW}}$, nicht aber mit $\overline{\text{MEMR}}$ und $\overline{\text{MEMW}}$ ansprechbar sein.

5.3.4 Anschluß der RAM/ROM-Karte an Ihren Amiga

Die Platine muß nach einer der vorgestellten Arten mit Speicher bestückt werden. Für einen ersten Test empfiehlt sich ein Betrieb als reine ROM-Disk nach Abschnitt 5.3.3.1. Beim Anschluß an den Rechner ist das Gerät unbedingt auszuschalten, ebenso eventuelle Zusätze.

Am einfachsten erfolgt die Installation beim Amiga 2000. Ziehen Sie dazu die Stecker für Tastatur und Maus an der Frontseite ab, und entfernen Sie den Gehäusedeckel, indem Sie ihn nach Lösen von fünf Schrauben leicht anheben und vorsichtig nach vorne abnehmen. Achten Sie darauf, daß er nicht an Kabeln hängenbleibt. Auf der linken Seite der Grundplatine kommen die Slots nach Bild 5.11 zum Vorschein. Drücken Sie die fertig bestückte RAM/ROM-Karte in einen freien der fünf 100-poligen Amiga-Slots. Die Bauteile müssen sich dabei auf der rechten Seite befinden, also zu Netzteil und Diskettenlaufwerken hinzeigen. Andersherum paßt die Karte ohnehin nicht in das Gehäuse. Achten Sie beim Schließen des Rechners darauf, daß keine Kabel eingeklemmt werden.

Für den Betrieb am Amiga 1000 oder am Amiga 500 muß die kleine Adapterplatine nach Bild 5.21 hergestellt werden. Sie enthält einen Slot-Stecker, in den die Platine dann wie beim Amiga 2000 eingedrückt werden kann. Leider sind zur Zeit 100-polige Slots nur schwer zu bekommen. Dagegen gibt es die kleineren PC-Slots überall. Wenn Sie zwei dieser Slots nach jeweils 50 Kontakten absägen, lassen sie sich sehr gut zu einem Amiga-Slot zusammensetzen. Für den Anschluß an den Rechner ist ein abgewinkelter 86-poliger Slot nötig, dessen Beschaffung zur Zeit leider auch nicht einfach ist. Dagegen werden normale Slots mit genügend langen Anschlüssen inzwischen von vielen Amiga-Lieferanten und von guten Elektronikläden angeboten. Das Abwinkeln muß man dann selbst vornehmen. Der Bestückungsplan in Bild 5.22 zeigt die genaue Lage der beschriebenen Bauteile. Hinweise zum Herstellen und Bestücken von Leiterplatten enthält Anhang A. Beginnen Sie beim Aufbau mit den 39 Durchkontaktierungen zwischen den Stekkern, dann bauen Sie den abgewinkelten Platinendirektstecker ein. Seine obere Kontaktreihe muß auf der Platine von oben verlötet werden. Nun folgt der Slot bzw. die beiden Hälften, und als letztes wird der Widerstand eingesetzt. Er dient als Pull-up für die Reset-Leitung, die bei manchen Ausführungen des Prozessors 68000 intern zu schwach geraten sind. Dieser Widerstand sollte jedoch nur einmal im System vorhanden sein.

Beim Einsatz mehrerer Erweiterungen kann er dort entfallen. In diesem Fall darf jedoch die Durchkontaktierung am Slot nicht vergessen werden.

Mit dem abgewinkelten 86-poligen Stecker passen die Platinen in den Systembus-Steckplatz, der beim Amiga 500 links und beim Amiga 1000 rechts an der Gehäuseseite hinter einer abnehmbaren Plastik-

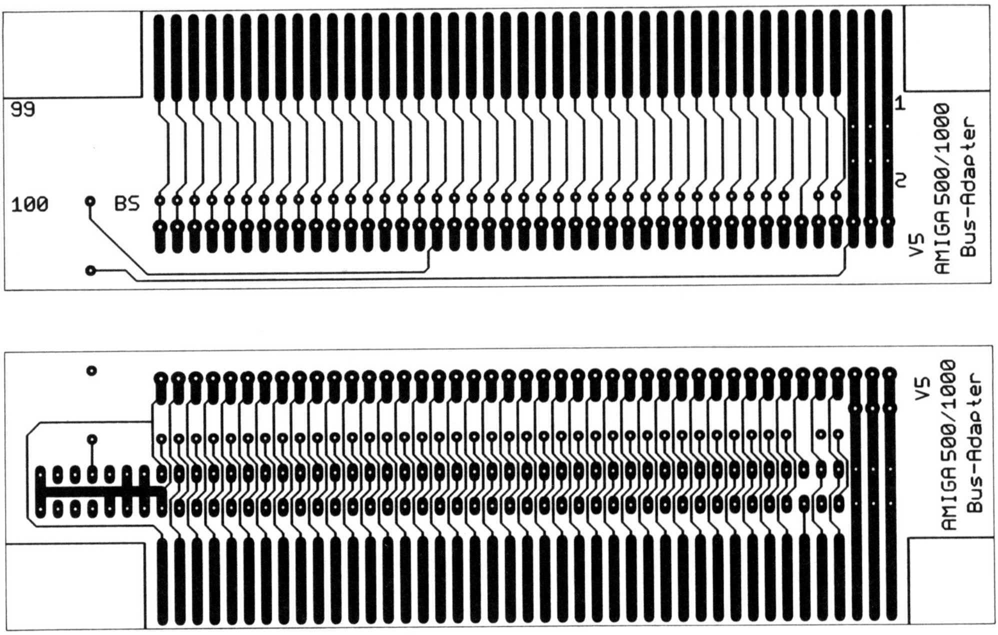


Bild 5.21: Bus-Adapter zum Anschluß einer Platine für den Amiga 2000 an den Amiga 500 oder 1000

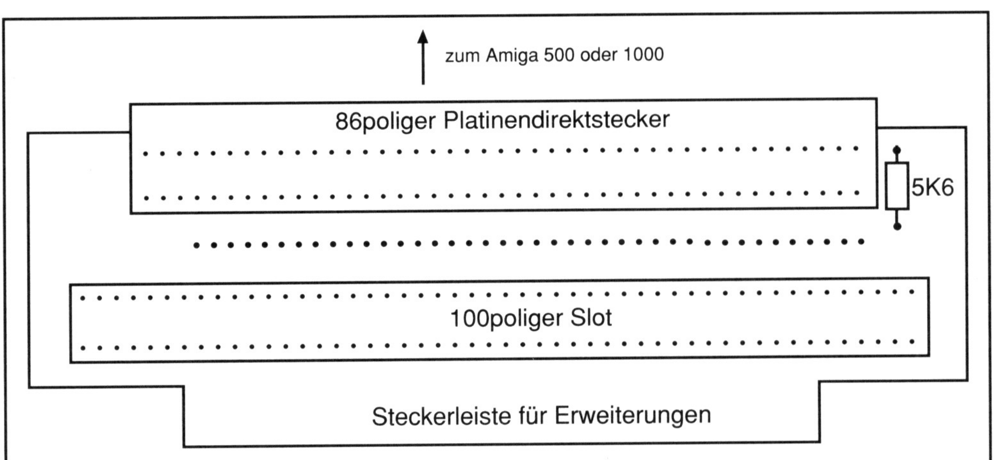


Bild 5.22: Bestückung des Bus-Adapters zum Anschluß an den Amiga 500 und 1000

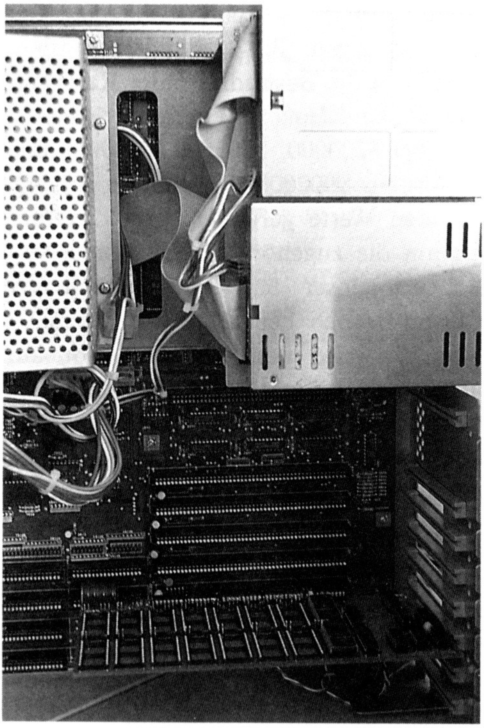


Foto 5.6: RAM/ROM-Karte im Amiga 2000

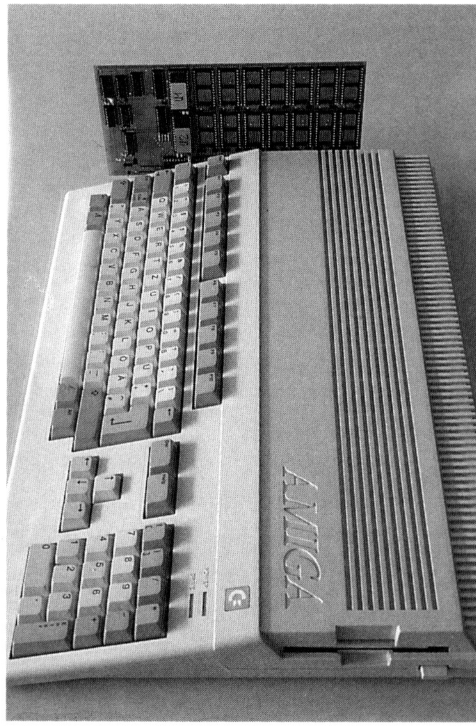


Foto 5.7: RAM/ROM-Karte im Amiga 500

abdeckung versteckt ist. Ihre genaue Lage zeigt Bild 5.11.

Der Adapter stellt die Bus-Steckverbindung durchgeschleift zur Verfügung, so daß gleichzeitig weitere Zusätze betrieben werden können. Allerdings sollten Sie nicht zu viele Schaltungen anschließen, da die Anordnung mechanisch zu wünschen übrig läßt, vor allem aber weil der Bus des Amiga 500 und 1000 nicht gepuffert ist und daher leicht überlastet werden kann.

Die Bauteilseite der in den 100-poligen Slot eingesteckten Zusatzplatine zeigt bei beiden Computern zum Rechnergehäuse hin.

5.3.5 Inbetriebnahme

Für einen ersten Funktionstest entfernen Sie vorsichtshalber alle Zusatzgeräte von Ihrem Amiga, die nicht unbedingt benötigt werden. Schalten Sie nun bei angeschlossener RAM/ROM-Karte den Rechner ein. Er muß reagieren wie gewohnt. Sollte das nicht der Fall sein oder sollten andere unvorhergesehene Dinge auftreten, schalten Sie ihn sofort wieder aus, entnehmen die Platine und kontrollieren sie nochmals intensiv anhand des Bestückungs- und des Schaltplanes auf Bestückungsfehler, kalte oder vergessene Lötstellen, Kurzschlüsse und ähnliches. Überprüfen Sie auch die Zuordnung der

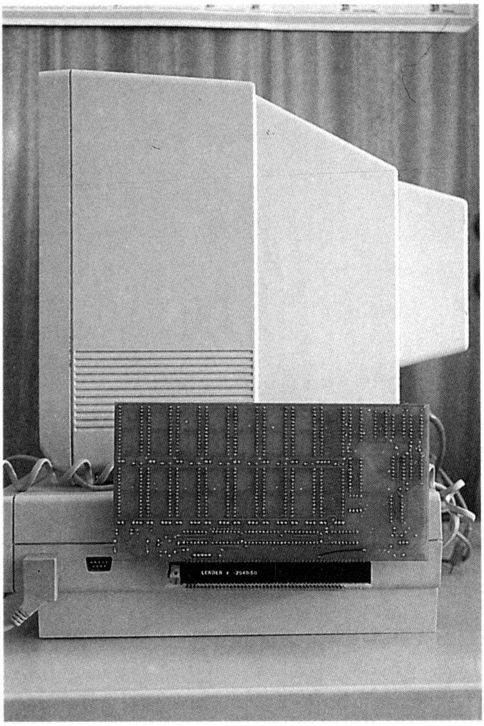


Foto 5.8: RAM/ROM-Karte im Amiga 1000

EPROMs zu den Sockeln, ihren Inhalt und die Position der Jumper. In jedem Fall müssen genau 24 Stück gesteckt sein. Ein Paar 2-Pin-Jumper bleibt immer leer!

Arbeitet Ihr Amiga normal, sollte die Karte bereits konfiguriert worden sein. Überprüfen Sie das durch Aufrufen des Programms RAMROMbase aus dem C-Directory der beiliegenden Diskette. Es sucht anhand der Produkt- und Herstellernummer die RAM/ROM-Karte im Speicher. Weil durch Beschreiben des Adreßregisters ausschließlich die vier höchstwertigen Bits veränderbar sind, kann sich die Basisadresse der Karte nur auf vollen Megabyte-Grenzen befinden, also bei $200000, $300000, $400000 und so fort bis $900000. Werden die entsprechenden Werte gefunden, gibt das Programm die zugehörige Basisadresse aus. Bei erfolgloser Suche wird auch der Config-Bereich bei $E80000 untersucht. Normalerweise liest der Prozessor auf unbelegten Adressen $FF. Dies sollte im Config-Bereich der Fall sein. Sind dort jedoch Informationen vorhanden, ist davon auszugehen, daß die Karte nicht konfiguriert wurde.

Handelt es sich um die vorgegebenen Daten, wird ausgegeben: »Board liegt noch im Config-Bereich ab $E80000!«. In diesem Fall ist besonders der Konfigurationsteil der Schaltung mit Demultiplexer U21, Register U25, Komparator U26 und Flipflops U23 zu überprüfen. Kann das Programm die Karte nicht erkennen, findet aber Daten im Config-Bereich, liegt wahrscheinlich ein Fehler in der Schreib/Lese-Logik vor oder die ROM-Bausteine wurden vertauscht. Überprüfen Sie besonders die Treiber U17..U20, Flipflops U24 und Demultiplexer U22 sowie die Position der Jumper. Auch die Power-Fail-Schaltung kann der Grund sein. Pin 6 von U22 muß im normalen Betrieb HIGH-Pegel führen. Schalten Sie den Rechner aus und entnehmen Sie die Karte. Wahrscheinlich haben Sie nur eine Kleinigkeit übersehen.

```
/************************************/
/*                                  */
/*            RAMROMbase            */
/*                                  */
/*     Ermittelt die Basisadresse   */
/*         des RAM/ROM-Boards       */
/*                                  */
/*       Uwe Gerlach,  01.02.89     */
/*                                  */
/************************************/

#include "stdio.h"

char  *base,                         /* Basisadresse für Boardsuche */
      found = 0;                     /* Flag für erfolgreiche Suche */
long  i;                             /* Zähler                      */

main()

{
for (base = 0x200000; base < 0xa00000; base += 0x100000)
  {
  if (((base[0x04] & 0xf0) == 0xd0) &&
      ((base[0x06] & 0xf0) == 0x40) &&   /* Produkt Nummer */
      ((base[0x10] & 0xf0) == 0x30) &&
      ((base[0x12] & 0xf0) == 0xf0) &&
      ((base[0x14] & 0xf0) == 0xf0) &&
      ((base[0x16] & 0xf0) == 0x70))     /* Herstellernummer */
    {
    found = 1;
    printf("\nBasisadresse RAM/ROM-Board: $%x\n\n", base);
    }
  }
if (found != 1)
  {
  base = 0xe80000;
  if (((base[0x04] & 0xf0) == 0xd0) &&
      ((base[0x06] & 0xf0) == 0x40) &&   /* Produkt Nummer */
      ((base[0x10] & 0xf0) == 0x30) &&
      ((base[0x12] & 0xf0) == 0xf0) &&
      ((base[0x14] & 0xf0) == 0xf0) &&
      ((base[0x16] & 0xf0) == 0x70))     /* Herstellernummer */
    {
```

```
      found = 1;
      printf("\nRAM/ROM-Board liegt noch im Config-Bereich ab $E80000!");
      }
    else
      {
      printf("\nBoard nicht erkannt!");
      for (i = 0; ((i < 80000) & (found == 0)); i++)
        {
        if ((base[i] & 0xff) != 0xff)
          found = 1;
        }
      if (found == 1)
        printf("Aber da ist was im Config-Bereich $E80000");
      }
    printf("\n\n");
    }
}
```

Listing 5.1: Das Programm RAMROMbase

Wird vom Programm RAMROMbase eine Basisadresse angezeigt, spricht das für die Funktionsfähigkeit der Karte. Unter Kickstart 1.3 und bei autobootfähigem RAM/ROM-Device muß die ROM-Disk und die eventuell vorhandene RAM-Disk dann bereits vom System installiert sein. In diesem Fall können Sie das folgende Kapitel überspringen.

5.3.5.1 Mounten unter Kickstart 1.2

Unter Kickstart 1.2 oder bei fehlender Autobootfähigkeit müssen Sie noch das Memory-Device installieren. Dazu werden die Daten des neuen Device DM0: in der Mountlist im Ordner Devs: benötigt. Die Angaben haben folgendes Format:

```
DM0:  Device = memory.device        /* Name des Device */
      Unit = 0                      /* logisches Laufwerk */
      Flags = 0                     /* Flags für OpenDev() */
      Surfaces =                    /* Anzahl Köpfe */
      BlocksPerTrack =              /* Sektoren pro Spur */
      Reserved = 2                  /* Reserviert für Bootblock */
      Interleave = 0                /* kein Spurversatz */
      LowCyl = 0                    /* niedrigste Spur */
      HighCyl =                     /* höchste Spur */
      Buffers = 5                   /* Pufferanzahl */
      BufMemType = 1                /* nutzt Chip- oder Fast-RAM */
#
```

```
DM1:  Device = memory.device      /* Name des Device */
      Unit = 1                    /* logisches Laufwerk */
      Flags = 0                   /* Flags für OpenDev() */
      Surfaces = 4                /* Anzahl Köpfe */
      BlocksPerTrack = 32         /* Sektoren pro Spur */
      Reserved = 2                /* Reserviert für Bootblock */
      Interleave = 0              /* kein Spurversatz */
      LowCyl = 0                  /* niedrigste Spur */
      HighCyl = 1                 /* höchste Spur */
      Buffers = 5                 /* Pufferanzahl */
      BufMemType = 1              /* nutzt Chip- oder Fast-RAM */
#
```

Je nach Bestückung der RAM/ROM-Platine müssen die Daten für Surfaces (normalerweise Anzahl der Köpfe), BlocksPerTrack (Sektoren pro Spur) und HighCyl (höchste erreichbare Spur) geändert werden. Bei der Zusammenstellung des ROM-Inhalts mit dem Programm MakeRomDisk werden die entsprechenden Daten abschließend ausgegeben.

Bei der RAM-Disk ist Surfaces immer 4 und BlocksPerTrack immer 32. Als HighCyl ist die Anzahl der vorhandenen RAM-Bänke vermindert um 1 anzugeben. Im Beispiel wurde eine Mountlist für vier RAM-Bausteine vom Typ 62256, also mit zwei Bänken abgedruckt (HighCyl = 1).

Fügen Sie also die entsprechenden Angaben in das File Devs:Mountlist Ihres Systems ein. Sie können die Beispieldatei auch von der beiliegenden Diskette aus Devs/DM-Mountlist zum Beispiel mit dem CLI-Befehl Join in Ihre Mountlist kopieren. Danach wird mit

```
Mount DM0:
Dir DM0:
```

die ROM-Disk DM0: angemeldet und installiert. Ebenso muß das bei der eventuell vorhandenen RAM-Disk DM1: geschehen.

5.3.5.2 Nutzung der RAM- und ROM-Disk

Egal ob nach Autoboot oder nach der Installation durch Mount. Schauen Sie sich doch einmal mit

```
Info
```

die Liste der zur Verfügung stehenden Laufwerke an. Dort sollte nun der Name DM0: mit der Zugriffsart »Write Onely« auftauchen. Das ist Ihre neue ROM-Disk. Bei vorhandenen RAMs erscheint auch die RAM-Disk DM1:. Sie muß jedoch – wie jede andere Diskette – vor der ersten Benutzung noch formatiert werden. Dies geschieht etwa mit

```
Format drive DM1: name CMOS-RAM
```

Nun läßt sie sich als normales Laufwerk benutzen. Wurden die CMOS-RAMs akugepuffert, bleibt die Formatierung und

der Inhalt auch nach dem Ausschalten erhalten. Die beiden virtuellen Laufwerke lassen sich genau wie die RAM-Disk RAM: des Systems verwenden. Mit einer Ausname: Natürlich kann auf die ROM-Disk nicht geschrieben werden. Bei der Arbeit mit der Workbench erscheint rechts für jedes angemeldete Laufwerk ein Disk-Icon.

Es kann wie üblich durch Hinzukopieren eines Files Disk.info abgeändert werden. Die beiliegende Diskette enthält unter ROM.info und RAM.info Beispiele.

Beim Anklicken erscheint ein Fenster, in dem bei entsprechendem Inhalt Programm-Icons erscheinen. Bei der RAM-Disk DM1: lassen sich auch andere Icons im Fenster ablegen und später wieder aufrufen.

Werden die CMOS-RAMs akkugepuffert betrieben, bleiben die einmal abgelegten Informationen wie bei einer ROM-Disk auch nach dem Ausschalten erhalten.

Auf der Karte lassen sich zum Beispiel die wichtigsten Befehle aus dem C-Directory und andere System-Dateien unterbringen. Stehen etwa die gebräuchlichsten Befehle im Ordner C auf DM0:, dann kann dies mit

```
Assign C: DM0:C
```

dem System mitgeteilt werden. Es holt sich dann diese Befehle bei Bedarf von dort. Sie müssen also nicht bei jedem Befehl die Bootdiskette einlegen und erreichen sogar eine höhere Ausführungsgeschwindigkeit als von einer Hard-Disk.

5.3.5.3 Booten von der RAM/ROM-Karte

Das Starten des Systems nach dem Einschalten ohne Einsatz einer Boot-Diskette bedeutet nicht nur eine Arbeitserleichterung, sondern bringt auch einen enormen Geschwindigkeitsvorteil mit sich. Mit der RAM/ROM-Karte ist das Booten wahlweise von der ROM-Disk DM0: oder von der RAM-Disk DM1: aus möglich. Die Auswahl geschieht über Boot-Prioritäten.

Im Kapitel 5.2.5 war bereits davon die Rede. Der zulässige Bereich ist –128 bis +127. Für die ROM-Disk wurde die Boot-Priorität 0 festgelegt. Befindet sich keine Karte im System, die eine höhere Priorität besitzt, und ist keine bootbare Diskette in Laufwerk DF0: eingelegt, wird das System von der ROM-Disk DM0: aus hochgefahren. Um wahlweise auch von der RAM-Disk DM1: booten zu können, muß deren Priorität erhöht werden. Das geschieht mit dem Programm SetBootPri. Als Argument muß die neue Priorität angegeben werden.

```
SetBootPri 1
```

etwa erhöht die Bootpriorität der RAM-Disk DM1: auf 1. Damit hat sie Vorrang vor der ROM-Disk DM0:. Die Priorität sollte jedoch immer kleiner als 5 gewählt werden, denn dies ist die vorgegebene Prioritätsstufe des Laufwerks DF0:. Die Boot-Priorität wird in einem der reservierten Blöcke der RAM-Disk gespeichert.

Nach einem Formatieren oder nach dem Befehl INSTALL muß sie daher erneut festgelegt werden. Wie bereits erläutert, ist Autoboot erst ab der Kickstart-Version 1.3 möglich. Die ROM-Disk muß mit

MakeRomDisk Scriptfile –A zusammengestellt worden sein.

Damit das System nach dem Einschalten von der RAM- oder ROM-Disk hochgefahren werden kann, sind wie bei jeder Bootdiskette die benötigten Systemverzeichnisse und -files dort abzulegen. Ein Beispiel für die Dateistruktur wäre:

```
S (dir)
Startup-Sequence

C (dir)
Mount
Assign
Execute
MakeDir

Devs (dir)
System-Configuration
Mountlist
harddisk.device

L (dir)
FastFileSystem
```

Im Verzeichnis S befindet sich die Startup-Sequence mit folgendem Inhalt:

```
Mount DH0:
Assign C: DH0:C
Assign Devs: DH0:Devs
Assign Libs: DH0:Libs
Assign Fonts: DH0:Fonts
Assign S: DH0:s
Assign L: DH0:l
Assign SYS: DH0:
MakeDir RAM:T
Assign T: RAM:T
Execute S:HD-Startup
```

Nach dem Einschalten konfiguriert der Amiga die RAM/ROM-Karte und installiert die RAM/ROM-Disk. Danach überprüft er, ob im Laufwerk DF0: eine bootbare Diskette liegt. Ist das nicht der Fall, lädt er aus dem RAM/ROM-Verzeichnis Devs das File System-Configuration und nimmt die Grundeinstellungen, wie Bildfarbe, Screenposition Mausparameter und so weiter, anhand dieser Daten vor. Danach wird aus dem Verzeichnis S die oben aufgeführte Startup-Sequence geladen, woraus die weiteren Aktionen hervorgehen. Die entsprechenden Befehle müssen im Verzeichnis C enthalten sein, wie auch die übrigen benötigten Dateien am jeweils geforderten Platz.

Im abgedruckten Beispiel wird die Harddisk angemeldet (Mount DH0:). Dazu wird das File Devs/Mountlist nach dem zugehörigen Eintrag durchsucht, L/FastFileSystem geladen und der Treiber Devs/harddisk.device installiert. Danach werden die Systemverzeichnisse mit Assign auf die Platte gelegt. Bevor mit Execute die Kontrolle an HD-Startup auf der Platte übergeben wird, erfolgt noch das Anlegen eines Verzeichnisses T für temporäre Dateien im RAM, denn der Befehl EXECUTE benötigt dies.

Dieses Beispiel soll nur als Anregung dienen und muß selbstverständlich an den persönlichen Bedarf angeglichen werden.

Anhang A

A.1 Praktische Tips zum Aufbau der Platinen

Bevor Sie mit dem Nachbau der beschriebenen Schaltungen beginnen, sollten Sie diesen Anhang unbedingt aufmerksam durchgelesen haben.

Er enthält Grundlagen zum Umgang mit elektronischen Elementen, wertvolle Tips für die Platinenherstellung, eine Einführung ins Löten sowie allgemeine Beschreibungen der wichtigsten verwendeten Bauteile samt ihrer Pinbelegung.

A.1.1 Der Arbeitsplatz

Zunächst brauchen Sie einen Tisch mit genügend Platz für die Materialien und zum Arbeiten, den Sie auch mal unaufgeräumt stehen lassen können, eine Steckdose in der Nähe und ausreichend Licht, weil sie es mit vielen kleinen Bauteilen zu tun bekommen werden.

Als Arbeitsunterlage kann eine alte Schreibtischmatte dienen, damit nichts Wertvolles durch herabfallende Lötzinntropfen angeschmort wird. Natürlich sollte auch der Computer nicht allzuweit weg stehen.

An dieser Stelle noch ein dringender Hinweis:

Die meisten Schaltungen in diesem Buch arbeiten mit so kleinen Spannungen, daß sie völlig ungefährlich sind.

Die wenigen Ausnahmen sind entsprechend gekennzeichnet.

Achten Sie in solchen Fällen besonders sorgfältig darauf, daß niemand eine offenliegende und von Ihnen unbewachte Schaltung berühren kann. Trennen Sie die Schaltung vorsichtshalber auch dann vom Netz, wenn Sie den Elektronik-Arbeitsplatz nur für wenige Augenblicke verlassen, oder noch besser, stecken Sie den Netzstecker nur in die Dose, wenn die entsprechende Schaltung ganz in ein sicher isolierendes Gehäuse eingebaut ist.

A.1.2 Das richtige Werkzeug

Elektronik-Arbeiten – gerade bei Zubehör für Computer – sind Feinarbeiten. In diesem Buch werden vorwiegend Platinen vorgestellt, die gar nicht erst ins Gehäuse eingebaut werden sollen, da sie an der Rückseite des Computers angesteckt werden, und dort auch so sicher genug sind. Wir können uns also im großen und gan-

zen auf das reine Elektronik-Werkzeug beschränken. Ein wunder Punkt ist allenfalls die Platinenherstellung, auf die wir gleich noch genauer zu sprechen kommen.

Wichtig ist eine Elektronik-Flachzange zum Abwinkeln der Anschlußdrähtchen vor dem Bestücken. Außerdem ein kleiner Saitenschneider, um die überstehenden Anschlußdrähtchen der Bauelemente abzuzwicken, nachdem sie in die Platine gesteckt wurden. Ein scharfes Bastelmesser und eine Abisolierzange helfen beispielsweise beim Vorbereiten von Litzen. Weiterhin sollte noch ein Schraubenzieher und eine Pinzette zur Verfügung stehen.

Wichtigstes Utensil ist natürlich der Lötkolben. Es ist ein Feinlötkolben mit dünner, verzinnter Spitze einzusetzen. Seine Leistung beträgt etwa 15 bis 30 Watt. Zu hohe Temperaturen können die empfindlichen Halbleiterbauelemente leicht zerstören.

Fast alle hochintegrierten ICs sind heute auf MOS-Basis gefertigt (Metall Oxide Semiconductors = Metalloxid-Halbleiter). Diese Bausteine nehmen Spannungsspitzen sehr ernst. Es genügt bereits eine kurze Schaltspitze auf der Netzleitung während des Lötvorgangs, um ein solches IC zu zerstören. Aus diesem Grunde sollte man sich besser einen Lötkolben mit Trenntrafo – auch Lötstation genannt – zulegen. Dort werden die gefährlichen Spitzen heruntertransformiert und können keinen Schaden mehr anrichten.

Neben dem eigentlichen Werkzeug braucht man noch einige Meßinstrumente, um die richtige Funktion eines Schaltungsteils überprüfen zu können. Hier sollten Sie mindestens über ein Vielfachmeßgerät verfügen. Es sollte einen Ohm-Meßbereich haben, um die Leiterplatten auf Lötbrücken bzw. Unterbrechungen kontrollieren zu können, sowie mindestens einen Spannungsmeßbereich und einen Strommeßbereich.

A.2 Platinen selbstgemacht

A.2.1 Vom Layout zur gedruckten Schaltung

In diesem Buch befinden sich Vorlagen zur Herstellung gedruckter Schaltungen. Sie haben oft dünne und dicht nebeneinanderliegende Bahnen und erfordern daher sorgfältige Behandlung. Grundmaterial zur Herstellung von gedruckten Schaltungen, wie Leiterplatten oft (nicht ganz richtig) genannt werden, ist immer ein isolierender Träger, der – je nach Anwendung – einseitig oder beidseitig mit Kupferfolie beschichtet ist. Zusätzlich ist eine Positiv-Fotolack-Schicht aufgebracht, die das Kupfer abdeckt.

Dort, wo diese Schicht belichtet wird, löst sie sich beim Entwickeln ab und gibt die Kupferschicht frei. Beim anschließenden Ätzen wird das Kupfer an den ungeschützten Stellen vom Träger abgelöst und übrig bleiben nur die im Layout schwarzen Leiterbahnen.

A.2.2 Bilder aus Kupfer

Erste Hürde bei der Platinenherstellung ist die richtige Belichtung. Machen Sie von den abgedruckten Vorlagen zwei Fotokopien auf Overheadfolie. Achten Sie

dabei darauf, daß der Maßstab genau 1:1 ist. Kleben Sie die beiden Folien mit Tesafilm so aufeinander, daß sich die Bahnen genau decken. Bei nur einer Folie sind die Bahnen nicht schwarz genug für eine Belichtung.

Die Platten können bedenkenlos bei gedämpftem Tageslicht verarbeitet werden. Nehmen Sie ein genügend großes Stück fotopositiv beschichtetes Basismaterial, ziehen Sie die schwarze Schutzfolie ab und legen Sie den gewünschten Ausschnitt der Layout-Folie so darauf, daß Sie die Schrift lesen können. Um sicherzustellen, daß die Folie wirklich plan aufliegt und sich nicht während der Belichtung durch die entstehende Wärme verzieht, wird sie mit einer sauberen Glasplatte beschwert. Das ganze beleuchten Sie etwa 6 bis 8 Minuten lang mit einer 500-Watt-Halogen-Kopierlampe, die im Abstand von etwa 25 Zentimetern genau darüber angebracht ist. Leichte Überbelichtung ist besser als Unterbelichtung, denn dabei wird später das entwickelte Bild nicht schleierfrei und sauber oder Schicht schwimmt fort.

Viele Hobby-Elektroniker sind bei doppelseitigen Platinen skeptisch. Das ist jedoch gar nicht nötig, wenn man einige Tricks beachtet. Zuerst wird an zwei angrenzende Ränder der Layout-Unterseite mit Tesafilm genau rechtwinklig je ein schmaler Abfallstreifen Basismaterial geklebt. Nun legt man den Oberteil des Layouts von der Gegenseite her auf die Streifen, hält das ganze gegen das Licht und bringt die zueinander gehörigen Lötaugen zur Deckung. Genau in dieser Lage wird nun auch die Oberseite des Layouts an die Streifen geklebt. So erhält man eine Tasche, in die das zu belichtende Material eingesteckt werden kann. Es sollte etwas größer sein als die Layouts, damit man es auf jeder Seite ebenfalls mit einem Streifen Tesafilm gegen Verrutschen sichern kann. Ober- und Unterseite werden jetzt nacheinander wie einseitige Platinen belichtet. Beim Herausnehmen bleibt die Tasche erhalten, so daß bei der nächsten Platine das erneute Justieren entfällt.

Sie werden sicherlich feststellen, daß die Lampe bei der Belichtung enorme Hitze entwickelt. Auch die der Lampe zugewandte Platinenseite erwärmt sich. So kommt es, daß bei doppelseitigen Platinen die zuletzt belichtete Seite wesentlich wärmer ist als die andere. Würde man das Material sofort entwickeln, geschähe das in unterschiedlicher Geschwindigkeit und die Aktion wäre zum Scheitern verurteilt.

Daher legen Sie die belichtete Platine zum Temperaturausgleich zunächst einmal mehrere Minuten bei gedämpftem Tageslicht oder bei Dunkelkammer-Rotlicht in Wasser.

A.2.3 Entwicklungshilfe

Nun kommt der kritischste Vorgang: das Entwickeln. Dazu wird die Platine in ein Natriumhydroxid-Bad getaucht, das auch als Ätznatron bekannt ist. Natriumhydroxid wird in Perlform geliefert und ist leicht in Wasser löslich, doch Achtung! Beim Lösen entstehen unangenehme Dämpfe und die Flüssigkeit wird erstaunlich warm. Vermeiden Sie unbedingt Berührungen mit dieser Chemikalie, ziehen Sie einen alten Kittel an und bedecken Sie den

Arbeitstisch mit einer dicken Lage Zeitungen. Die genaue Dosierung beim Lösen ist auf der Flasche angegeben und sollte unbedingt eingehalten werden.

Je nach Zustand der Lösung dauert der Entwicklungsvorgang meist nur wenige Sekunden. Sein Ende erkennt man daran, daß an den belichteten Stellen das blanke Kupfer zu sehen ist. Man darf die Platine jedoch nicht zu lange entwickeln, sonst wird der Lack auch dort abgelöst, wo er eigentlich stehenbleiben müßte. Zur Kontrolle nehmen Sie die Platine immer wieder aus der Flüssigkeit und spülen sie unter klarem Wasser ab. Wichtig ist, daß die Löcher in den Lötaugen zu erkennen sind, denn sie müssen später unbedingt freigeätzt werden. Fassen Sie jetzt nicht mehr auf die entwickelten Bahnen, sonst kann es passieren, daß an einigen Stellen das Kupfer durch anhaftendes Fett nicht sauber weggeätzt wird.

Dauert die Entwicklung zu lange, ist die Platine entweder unterbelichtet, oder die Entwicklerlösung zu schwach oder verbraucht. Geht sie zu schnell, ist dagegen die Lösung zu stark (mit Wasser verdünnen) oder das Bad zu warm (über 30 Grad).

A.2.4 Es wird ätzend

Inzwischen können Sie die Ätzlösung ansetzen. Das geht mit Eisen-III-Clorid oder mit Ätzsulfat. Beide Mittel werden als Granulat geliefert und erzeugen ihre volle Ätzkraft bei Temperaturen zwischen 45 und 50 Grad. Eisen-III-Chlorid besitzt dabei den Nachteil schnell nachlassender Ätzkraft, rotbrauner Färbung (schlechter Sichtkontakt) sowie der Entwicklung von unangenehmen Gerüchen und Dämpfen.

Ätzsulfat dagegen ergibt eine klare Lösung, die nicht riecht und keine giftigen Dämpfe erzeugt. Auch hier sollte man Kontakt mit Haut und Augen sowie Textilien vermeiden. Gegebenenfalls sofort mit lauwarmem Wasser und Seife abspülen.

Man kann das Granulat einfach in heißem Wasser lösen. Dann muß es jedoch vor dem Abkühlen verarbeitet werden. Profis verwenden Ätzanlagen mit Heizung, mit denen die Arbeit natürlich weit weniger aufwendig ist.

Sorgen Sie dafür, daß sich die Flüssigkeit ständig bewegt. Nach etwa 5 bis 10 Minuten ist der Ätzvorgang beendet. Das vorher blanke Kupfer muß restlos von der Trägerschicht abgelöst sein. Wichtig ist vor allem, daß keine Verbindungen zwischen einzelnen Leiterbahnen mehr stehenbleiben. Achten Sie auch darauf, daß die Mittelpunkte der Lötaugen deutlich weggeätzt sind. Sie dienen nämlich beim anschließenden Bohren als Zentrierung. Es kann nichts schaden, das Material etwas länger im Ätzbad zu lassen.

Ändert sich am Zustand nichts mehr, dann nehmen Sie die Platine aus dem Bad, spülen sie unter fließendem Wasser ab und trocknen sie mit einem saugfähigen Papier. Den restlichen Fotolack auf den Leiterbahnen sollte man zum Schutz vor Oxidation stehenlassen. Beim Löten brennt er sich an den erhitzten Stellen leicht fort und stört nicht.

Die benötigten Chemikalien können Sie mehrmals verwenden, bis die jeweilige

Lösung verbraucht ist, was sich durch deutlich langsamere Reaktionen bemerkbar macht. Verbrauchte Ätzlösungen dürfen erst nach geeigneter Entgiftung ins Abwasser gelangen. Hierzu wird die Ätzlösung auf etwa ihr achtfaches Volumen mit Wasser verdünnt und unter ständigem Rühren zehnprozentige Natronlauge (pro Liter Ätzlösung cirka ein Liter Natronlauge erforderlich) zudosiert, bis sich ein pH-Wert von 10 in der Lösung einstellt. Das kann man beispielsweise mit einem Indikatorstäbchen überprüfen. Es entsteht ein voluminöser Niederschlag (Metallhydroxide), der sich langsam absetzt. Nach entsprechender Absetzzeit kann die entgiftete Lösung abgefiltert werden.

A.2.5 Bohren und Bestücken

Nun müssen noch die Löcher für die Bauteile gebohrt werden. Dazu braucht man auf jeden Fall einen Bohrständer, da die feinen Bohrer sehr leicht abbrechen. Standard-Lochdurchmesser ist 0,8mm. Für größere Bauteile muß jedoch mit 1mm bzw. 1,3mm gebohrt werden. Diese Maße sollten möglichst eingehalten werden, damit sowohl elektrisch als auch mechanisch sichere Kontakte entstehen.

Nach dem Bohren wird eine Sichtkontrolle vorgenommen, indem man die Platine auf feine Leiterbahnunterbrechungen und Überbrückungen hin durchsieht. Unterbrechungen können mit einem kleinen Drahtstück oder einem Klecks Lötzinn geflickt und Kupferbrücken mit einem schmalen Schraubenzieher oder einer Reißnadel entfernt werden. Auch beim Bestücken gibt es einige Grundregeln zu beachten. Da ist zum einen die Reihenfolge. In der Regel beginnt man mit den niedrigsten Elementen, also mit Drahtbrücken, falls diese vorhanden sind, fährt dann mit den Widerständen fort und gelangt danach zu IC-Sockeln, Kondensatoren und so fort. Zum Schluß kommen die empfindlichen Halbleiter an die Reihe, wie Transistoren und ICs. Man sollte sich angewöhnen, alle ICs zu sockeln. Das kostet vielleicht ein paar Mark mehr, aber man tut sich um vieles leichter, wenn einmal ein Baustein ausgewechselt werden muß.

Achten Sie besonders auf die korrekte Einbaurichtung von Halbleitern und Elkos. Einige Seiten weiter finden Sie Anschlußbelegungen vieler Bauteile angegeben. Sollten Sie abweichende Gehäuseausführungen erwischt haben, dann fragen Sie unbedingt beim Händler nach den entsprechenden Daten. Falsch eingebaute Teile können nicht nur eine ordnungsgemäße Funktion verhindern, sondern führen oft unversehens auch zur Zerstörung anderer Bauelemente.

A.3 Die Kunst des Lötens

Der Lötdraht des Elektronikers hat mit dem Lötdraht des Bauklempners nur wenig gemeinsam. Der glänzende Zinnmantel enthält eine gelblich-braune Masse: die Kolophoniumeinlage. Auf diese Masse kommt es beim Löten besonders an, denn sie ist das Flußmittel, das während des Erhitzens die Lötstelle von Oxidschichten und Verunreinigungen säubern soll, damit ein mechanisch fester und gut leitender Kontakt entsteht.

Der Lötkolben muß vor Beginn der Arbeiten Betriebstemperatur haben. Man prüft

das, indem etwas Lötzinn auf die Spitze gehalten wird. Das Flußmittel sollte dabei sofort verdampfen und etwas Zinn auf der Spitze zerfließen.

Jedes Bauelement wird einzeln vorbereitet. Bei Widerständen geschieht das beispielsweise, indem die beiden Anschlußdrähtchen im richtigen Abstand rechtwinklig umgebogen werden. Man steckt anschließend die langen Drahtenden bis zum Anschlag durch die vorgesehenen Bohrungen und knickt sie von der Unterseite her leicht nach außen, damit das Element nicht wieder herausfallen kann. Beim Einlöten können Sie ja nichts mehr festhalten.

Was jetzt folgt ist eigentlich gar nicht schwer. Es hört sich nur kompliziert an. Wichtig ist in jedem Fall Erfahrung, die erst nach einigen Versuchen kommen kann. Verzweifeln Sie also nicht gleich.

Falls irgendwie möglich, lagern Sie die Lötstelle so, daß die Schwerkraft den Lötvorgang unterstützt. Es ist günstig, wenn das Zinn zur Lötstelle hinfließt und nicht von ihr weg. Sobald die Lötstelle ausreichend erwärmt ist, bringen Sie das Zinn an die Lötstelle heran. Dabei fließt zuerst das im Lötdraht eingebettete Kolophonium und dann erst das Zinn über die Lötstelle. Diese Reihenfolge ist vorgesehen.

Sobald genügend Zinn vom Lötdraht abgeschmolzen ist, nehmen Sie den Lötdraht von der Lötstelle weg. Wichtig ist, daß die Lötkolbenspitze gerade so lange an der Lötstelle gehalten wird, bis alle an der Lötung beteiligten Teile vom Zinn umflossen sind.

Wie bereits erwähnt, dürfen insbesondere Halbleiterbauelemente – also Transistoren, Dioden usw. – beim Einlöten nicht zu heiß werden. Die Lötzeit sollte daher möglichst kurz bleiben (maximal 5 Sekunden).

Die Lötstelle darf während des Erkaltens nicht mehr bewegt werden. Den Abkühlvorgang kann man beschleunigen, indem man den frischen Kontakt leicht anbläst. Eine fachgerechte Lötstelle glänzt silbrig und hat eine glatte Oberfläche.

Sollte beim Lötvorgang das Lot gut schmelzen, aber nicht ausreichend breit verlaufen, dann ist nicht genügend Flußmittel an die Lötstelle gelangt. Führen Sie noch etwas Lötdraht zu. Dasselbe tun Sie auch, wenn Sie eine erkaltete Lötstelle noch einmal bearbeiten wollen. Achten Sie aber darauf, daß nicht zuviel Lötzinn auf den Kontakt gelangt und eine unbeabsichtigte Lötbrücke zu anderen Leiterbahnen bildet. Zuviel Lötzinn und im Falle eines Falles auch eine Brücke kann mit einem Entlötgerät leicht abgesaugt werden.

Ist alles erfolgreich »verlaufen«, wird das überstehende Drahtende mit einem Saitenschneider abgeknipst.

So muß man nun nacheinander bei jeder Lötstelle vorgehen.

Wenn alles bestückt ist, sollte nocheinmal eine Sichtkontrolle durchgeführt werden, denn es kann vorkommen, daß sich haarfeine Lötbrücken gebildet haben. Das geschieht leicht bei eng aneinanderliegenden Lötstellen, oder bei ICs mit Leiterbahndurchführungen zwischen den An-

schlußbeinchen. Eine sorgfältige Kontrolle vor dem ersten Einschalten kann eine spätere, langwierige Fehlersuche ersparen, da Halbleiter leicht durch falsches Anschließen zerstört werden können, und dann unter Umständen gar nichts mehr geht.

A.4 Der unvermeidbare Kleinkram

Alle in den vorgestellten Schaltungen benötigten Bauteile kann man in Elektronik-Läden kaufen. Bei seltenen oder schwer beschaffbaren Teilen ist in der Bestückungsliste eine Bezugsadresse angegeben. Beachten Sie auch die Adressen von Versandgeschäften im Anhang C.

Nicht immer werden Sie im Geschäft um die Ecke genau das Teil erhalten, das in der Liste aufgeführt ist. Wollen Sie zum Beispiel einen ganz bestimmten Transistor kaufen, kann es durchaus vorkommen, daß Sie gefragt werden, ob es auch ein Universaltyp sein kann, oder wenn Sie einen Widerstand mit 340 Ohm verlangen, kann es heißen: »Können es auch 330 Ohm sein?«.

Dazu muß man wissen, daß elektronische Bauteile herstellungsbedingt immer eine gewisse Toleranz aufweisen. Ein Widerstand mit 330 Ohm und 10% Toleranz – dieser Wert ist üblich – kann damit Werte zwischen 297 und 363 Ohm haben. Es wäre für die Händler nicht zumutbar, jedem Ohmwert eine eigene Schublade zuzuordnen. Außerdem kommt es in der Praxis nur bei sehr wenigen Anwendungen auf exakte Werte an. Gerade in der Digitaltechnik sind die Ansprüche meist äußerst gering, da normalerweise alle für die sichere Funktion verantwortlichen Teile bereits in den ICs untergebracht sind. Pull-up-Widerstände sind beispielsweise völlig unkritisch. Ihr Wert kann durchaus zwischen etwa 400 Ohm bis 10000 Ohm schwanken.

Die Bauteile werden daher in Normreihen eingeteilt, die alle möglichen Werte abdecken. Der Nennwert des nächsthöheren Widerstandswertes in der Normreihe E12 (12 Werte pro Dekade) mit ±10% Toleranz ist zum Beispiel 390 Ohm. Suchen Sie sich also ruhig den nächstliegenden vorhandenen Wert aus.

Viele Baugruppen sind bereits mit aufeinander abgestimmten Einzelelementen in ein gemeinsames Gehäuse eingebaut. Musterbeispiel dafür ist natürlich das IC. Beinhaltet ein Gehäuse dagegen tatsächlich nur ein einziges Bauelement im elektrischen Sinn, dann spricht man von einem diskreten Bauelement. Im Folgenden einige wichtige Angaben zu solchen universellen Elementen.

A.4.1 Widerstände

Wie Sie bereits wissen, werden Widerstände in Ohm (Ω) gemessen. Statt 1000 Ohm sagt man auch Kiloohm (kΩ) und statt 1 000 000 Ohm Megaohm (MΩ). Um verschiedene Werte unterscheiden zu können, kennzeichnet man sie mit vier oder fünf Farbringen. Bild A.1 zeigt deren Bedeutung. Man liest die Ringe von demjenigen Anschluß her, dem sie am nächsten liegen. Metallfilmwiderstände, die in der Regel mit geringeren Toleranzen als Kohleschichtwiderstände hergestellt werden,

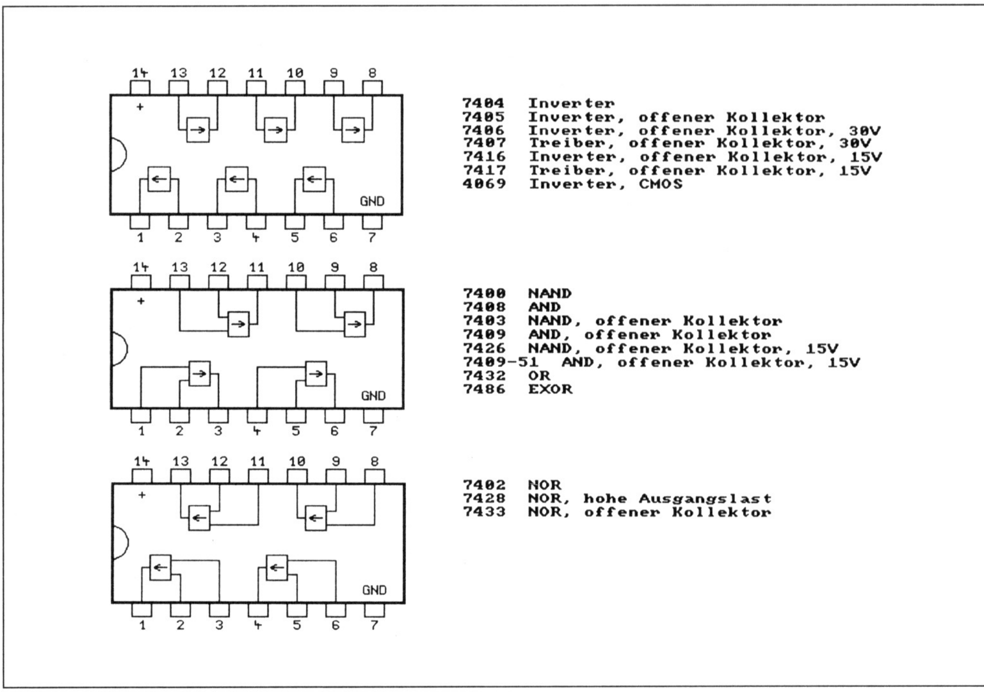


Bild A1: Farbcode-Tabellen für Schichtwiderstände (IEC-Norm und DIN 41 429)

tragen fünf Ringe. Damit lassen sich genauere Wertangaben machen.

Das Symbol für einen Widerstand ist ein Rechteck. In Schaltbildern, Bestückungsplänen und Bauteillisten werden Widerstandswerte oft abgekürzt. So hat es sich eingebürgert, das Zeichen für Ohm (Ω) einfach wegzulassen. Außerdem wird an die Stelle des Kommas häufig der Multiplikator k oder M gesetzt. Ein Widerstand von 5600 Ohm erscheint also zum Beispiel als 5,6k oder auch als 5k6.

1M5 wäre ein Widerstand mit 1,5 Megaohm – das sind 1 500 000 Ohm.

Widerstände gibt es nicht nur mit verschiedenen Ohm-Werten, sondern auch für unterschiedliche Belastungen. In jedem Widerstand wird elektrische Leistung in Wärme umgewandelt. Je mehr Strom fließt, bzw. je höher die Spannung steigt, desto wärmer wird es dem Bauelement. Irgendwann aber haucht auch der eifrigste Widerstand sein Leben in Form von Rauchzeichen aus. Die dafür nötige Leistung nennt man die maximale Verlustleistung. Für unsere Schaltungen genügen Widerstände mit 1/4 Watt Belastbarkeit, falls nicht anders gekennzeichnet. Natürlich können auch kräftigere Ausführungen gewählt werden. Diese sind dann allerdings teurer und brauchen mehr Platz.

Außer festen Widerständen gibt es noch beliebig einstellbare, die sogenannten Potentiometer oder einfach Potis. Sie lassen sich zwischen 0 Ohm und dem auf-

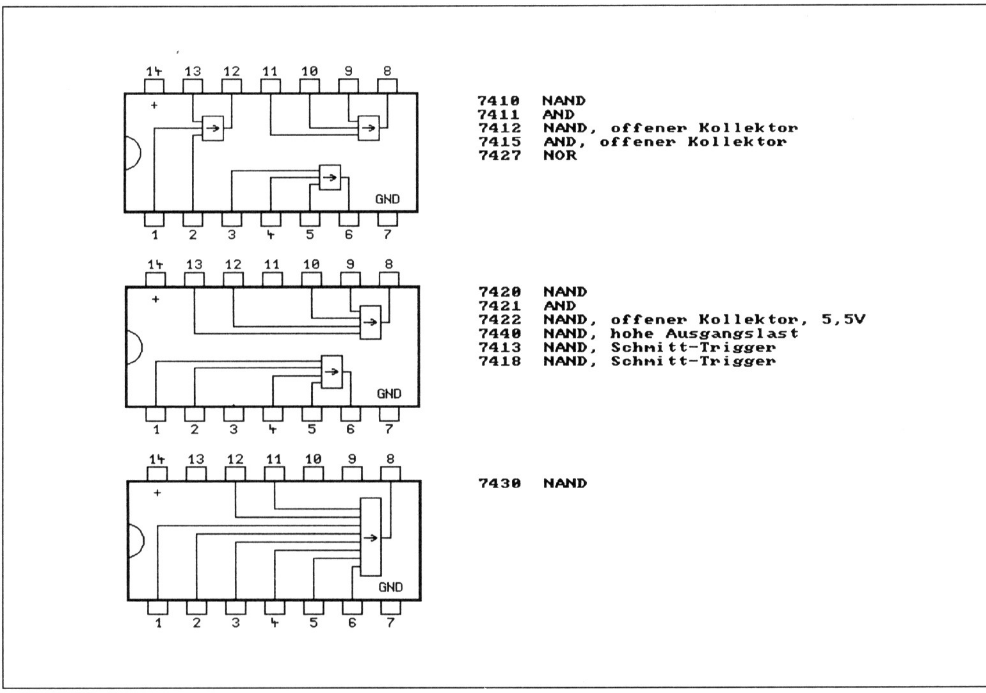


Bild A.2: Schaltsymbol eines Kondensators

gedruckten Maximalwert regeln. Trimmer sind ebenso aufgebaut, lassen sich aber nur mit einem Schraubenzieher verstellen.

A.4.2 Kondensatoren

Kondensatoren sind vom Prinzip her ganz einfache Bauelemente: zwei leitende Flächen, dazwischen eine Isolierschicht. Es gibt jedoch viele verschiedene Bauformen. In den meisten Anwendungen ist der Typ nicht maßgebend, so daß die Auswahl lediglich von den Kosten bestimmt wird. Die Kapazität eines Kondensators wird in Farad (F) angegeben. Da jedoch 1F bereits ein ungewöhnlich hoher Wert ist, sind die Bezeichnungen μF (Mikrofarad = 0,000 1 F), nF (Nanofarad = 0,000 000 1 F) und pF (Pikofarad = 0,000 000 000 1 F) üblich.

Nach der Art ihres Aufbaus ist eine Einteilung in Folienkondensatoren, Keramikkondensatoren und Elektrolytkondensatoren möglich. Hier ein paar Hinweise auf besondere Merkmale und Unterschiede:

Folienkondensatoren werden aus mit Metallschichten bedampften Folien gewickelt. Es lassen sich sehr genaue Werte mit geringen Temperaturschwankungen erzielen (zum Beispiel Styroflex-Kondensatoren). Einige Ausführungen eignen sich jedoch nicht für Hochfrequenzanwendungen.

Keramik-Kondensatoren werden dagegen

hauptsächlich in der Hochfrequenztechnik eingesetzt.

Auch für Digitalschaltungen sind sie zum Abblocken von sehr kurzen Spannungsspitzen besonders geeignet, um Störungen von der Betriebsspannung fernzuhalten. Kurzzeitige Spannungsschwankungen werden sozusagen durch die im Kondensator gespeicherte Ladung aufgefangen.

Solche Kondensatoren werden hauptsächlich im Pico- und Nanofaradbereich gebaut.

Elektrolytkondensatoren setzt man vorzugsweise bei sehr hohen Kapazitätswerten ein. Sie zeichnen sich durch eine relativ große Kapazität bei vergleichsweise geringer Baugröße und niedrigem Preis aus. Im Gegensatz zu den beiden anderen Bauformen sind »Elkos« jedoch gepolt und dürfen nicht an Wechselspannung betrieben werden. Bei falschem Einbau in die Platine wird ihre Isolierschicht zersetzt, was zu Kurzschluß und sogar zur Explosion führen kann! Die Polung des Bauteils ist daher in jedem Fall deutlich lesbar aufgedruckt. Bild A.2 zeigt die unterschiedliche Darstellungsweise eines gepolten und eines ungepolten Kondensators im Schaltplan. Der Pluspol wird durch ein Rechteck symbolisiert.

Bild A.3 zeigt die beiden wichtigsten Bauformen von Elkos. Unsere Platinen sind aus Platzgründen in der Regel für radiale Ausführungen vorgesehen. Bei axialen Typen ist der Minuspol zusätzlich mit einem umlaufenden Strich, der Pluspol dagegen mit einer ebensolchen Kerbe gekennzeichnet.

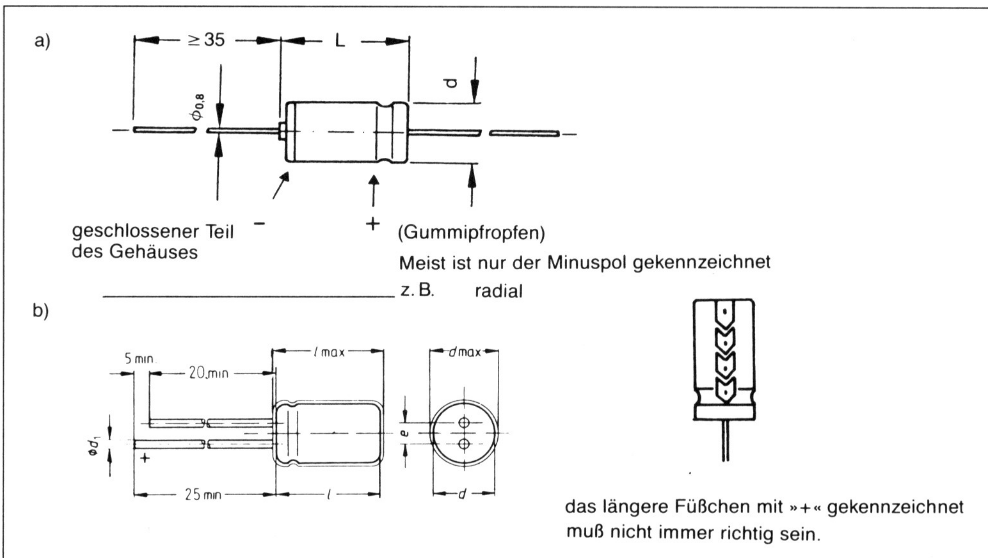


Bild A.3: Bauformen von axialen (a) und radialen (b) Elkos mit ihrer Polaritätskennzeichnung

Elektrolytkondensatoren haben sehr große Toleranzen und verändern ihre Werte sogar mit der Zeit und mit den Betriebsbedingungen.

Im Vergleich zu den am häufigsten verwendeten Aluminium-Elektrolytkondensatoren besitzen die tropfenförmigen Tantal-Elektrolyt-Kondensatoren bessere elektrische Eigenschaften und noch kleineren Raumbedarf. Sie sind jedoch ungleich teurer.

Die Kapazitätsangabe bei Kondensatoren ist mitunter etwas merkwürdig, und man muß schon ein wenig Phantasie besitzen und ungefähr den Wert abschätzen können, um sichere Aussagen machen zu können. Der Aufdruck .01 auf einem keramischen Scheibenkondensator bedeutet zum Beispiel 0,01 Mikrofarad, die Angabe 100 auf einem kleinen Keramik- oder Kunststofftyp dagegen 100 Pikofarad. 10/6 auf einem Elko heißt 10 Mikrofarad und maximal 6 Volt. Besonders auf japanischen Kondensatoren kann man alle möglichen Buchstaben sehen. Man sollte das nicht zu ernst nehmen. MF heißt nicht etwa Millifarad oder gar Megafarad (!), sondern es sind Mikrofarad gemeint. Oft sieht man dafür auch das Kürzel UF.

Jeder Kondensator ist für eine bestimmte maximale Spannung gebaut, die nicht überschritten werden darf. Andernfalls kann das Isoliermaterial zwischen den beiden leitenden Schichten zerstört werden. Diese maximale Spannung ist meist auf den Bauelementen angegeben. Sie liegt bei Folien- und Keramikkondensatoren so hoch, daß wir uns bei ihnen keine Gedanken machen müssen. Anders jedoch bei Elektrolytkondensatoren, die sich – je nach Spannungsfestigkeit – recht deutlich in Baugröße und Preis unterscheiden. Grundsätzlich kann man selbstverständlich Kondensatoren mit größeren Spannungsangaben als gefordert in die Schaltungen einbauen, man muß jedoch darauf achten, ob genügend Raum zum Einlöten auf der Platine vorhanden ist.

A.4.3 Halbleiter

Unter diesen Begriff fallen sehr viele unterschiedliche Bauelemente aus Halbleitermaterial (Germanium, Silizium...), zum Beispiel:

- Gleichrichterdioden
- Zenerdioden
- Fotodioden
- Leuchtdioden
- Kapazitätsdioden
- Transistoren
- Fototransistoren
- Operationsverstärker
- Brückengleichrichter
- Integrierte Schaltungen (ICs)
- Thyristoren
- Triacs
- Diacs
- Meßfühler (NTC, PTC) usw.

Außer bei manchen Meßfühlern handelt es sich immer um gepolte Bauelemente, die bei falschem Einbau zerstört werden können. Achten Sie daher unbedingt auf die richtige Einbaulage nach Bestükkungsplan. Die Bezeichnung ist bei jedem Bauteil aufgedruckt. Angaben über die jeweilige Anschlußbelegung entnehmen Sie bitte den zugehörigen Skizzen.

Für Einzelhalbleiter – auch diskrete Halbleiter genannt – gibt es umfangreiche Vergleichslisten. Bei den hier vorgestellten Schaltungen darf man ruhig darin aufgeführte entsprechende Vergleichstypen verwenden, falls nicht ausführlich davon abgeraten wird.

A.4.3.1 Transistoren

In der Digitaltechnik setzt man Transistoren vorrangig als Schalter ein, die durch eine elektrische Steuerspannung geöffnet und geschlossen werden. Der Schaltkontakt liegt zwischen Kollektor und Emitter, während die Steuerspannung an die Basis angelegt wird. Es genügt bereits ein kleiner Basisstrom, um den Schalter zu steuern. Dieser Sachverhalt wird in vielen Schaltungen besonders in Kapitel 2 ausgenutzt.

Grundsätzlich gibt es zwei Arten von Transistoren, die nach ihrem inneren Aufbau in NPN- und PNP-Typen aufgeteilt werden. Ihre Unterschiede erläutert Kapitel 2.3.4.

A.4.3.2 Leuchtdioden

Dioden sind zweipolige Halbleiterbauelemente, die nur einen Strom fließen lassen, wenn die anliegende Spannung in Richtung des stilisierten Pfeils in ihrem Schaltsymbol kleiner ist als am anderen Anschluß, sonst aber sperren. Bild A.4 zeigt das Schaltbild einer Diode und die Kennzeichnung der Kathodenseite am Gehäuse.

Leuchtdioden (LEDs = Light Emitting Diods) haben ähnliche Eigenschaften. Allerdings wird der P-N-Übergang so gestaltet, daß er bei Stromfluß Photonen aussendet, also leuchtet. Das zur Herstellung verwendete Material bestimmt die Wellenlänge, was gleichbedeutend ist mit der Leuchtfarbe. Das Spektrum reicht dabei vom unsichtbaren infrarot über rot, gelb und grün bis blau. Dabei nimmt der Wirkungsgrad jedoch stetig ab und blauleuchtende LEDs sind zudem sehr teuer.

Die wichtigsten Vorteile von Leuchtdioden gegenüber Glühbirnchen sind der wesentlich geringere Leistungsbedarf, die ungleich längere Lebensdauer, die Unempfindlichkeit gegenüber mechanischen Stößen und vor allem die kurze Reaktionszeit beim Ein- und Ausschalten. Nach dem Einschalten der Spannung leuchtet eine LED bereits nach wenigen tausendstel Teilen einer millionstel Sekunde mit ihrer vollen Lichtstärke.

Leuchtdioden haben normalerweise zylindrische Plastikgehäuse nach Bild A.5 mit 3mm oder 5mm Durchmesser. Zur Kennzeichnung der Polung ist das Gehäuse an

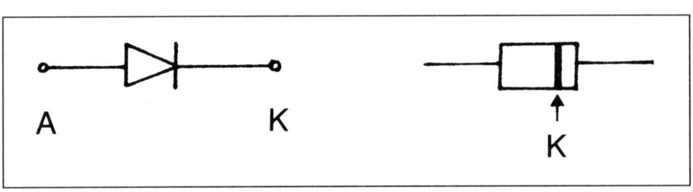


Bild A.4: Anschlußbelegung von Dioden (z.B. IN4148, IN4004...)

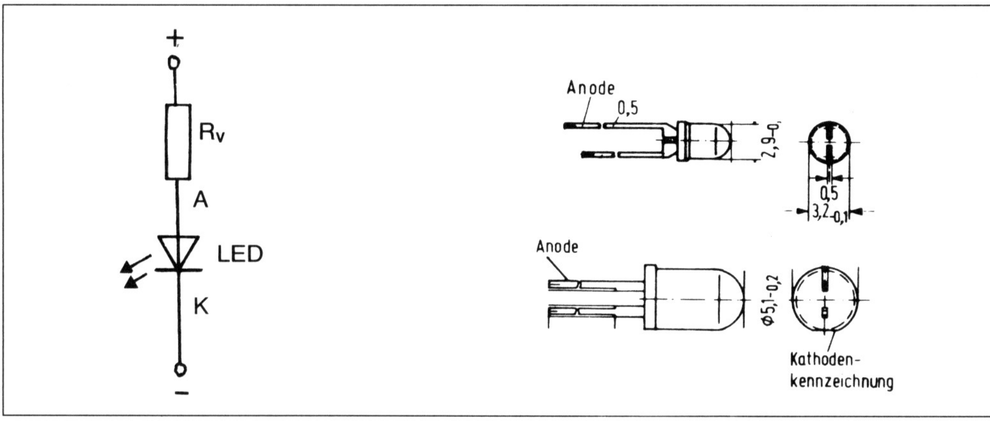


Bild A.5: Beschaltung und Anschlußbelegung von Leuchtdioden

der Kathodenseite meist abgeflacht. Üblicherweise hat die Anode ein längeres Anschlußdrähtchen.

Eine Leuchtdiode darf nie ohne Vorwiderstand an die Spannung geschaltet werden, der den Strom begrenzt und damit die Helligkeit der LED einstellt. Bei 5 Volt sind Werte zwischen 330 Ohm und 1 Kiloohm üblich.

A.4.3.3 Die Behandlung von MOS-Bauteilen

MOS steht für den englischen Begriff Metall Oxide Semiconductor – zu deutsch: Metalloxid-Halbleiter. Gemeint ist eine bestimmte Fertigungsweise von elektronischen Halbleiterbauelementen. Diese Bauteile sind sehr empfindlich gegen statische Aufladung und falsches Einsetzen in die Schaltung. Statische Ladung kann zum Beispiel entstehen, wenn Sie über einen Teppichboden gehen oder Kleidung mit Kunstfasern tragen. Durch Reibung laden Sie sich gegenüber Ihrer Umwelt elektrisch auf. Die so entstehende elektrische Spannung kann sehr hohe Werte annehmen. Berühren Sie nun ein MOS-IC, so wird dies mit hoher Wahrscheinlichkeit zerstört.

Deshalb sind beim Umgang mit MOS-Bauteilen folgende Vorsichtsmaßnahmen notwendig:

1. Sorgen Sie dafür, daß Ihr Körper nicht statisch geladen ist. Die statische Körperladung kann man durch Abreiben der Hände mit einem leitenden Material reduzieren.

2. Vermeiden Sie jeden direkten Kontakt mit den Anschlußpins, auch mit statisch aufladbarem Material, zum Beispiel Nylon. Dazu gehört auch die Lagerung der ICs in Kästchen aus Kunststoffmaterial.

3. Für den Transport sollten die Anschlußpins in einem leitenden Schaumstoff stecken, es sei denn, sie befinden sich bereits auf der Platine. Nur diese beiden Transportbedingungen halten die stati-

schen Ladungen von den Anschlußpins fern.

4. Beim Einstecken der ICs in die Fassungen muß man auf ihre richtige Lage achten. Kennzeichen dafür ist die IC-Kerbe, wie bereits in Kapitel 1 erläutert.

A.4.4 Andere Bauelemente

Es gibt in der Elektronik natürlich noch eine Vielzahl anderer Elemente. Alle aufzuzählen, würde bei weitem den Rahmen dieses Buches sprengen. Spezielle Teile werden jeweils im Text der zugehörigen Bauanleitung genauer erklärt.

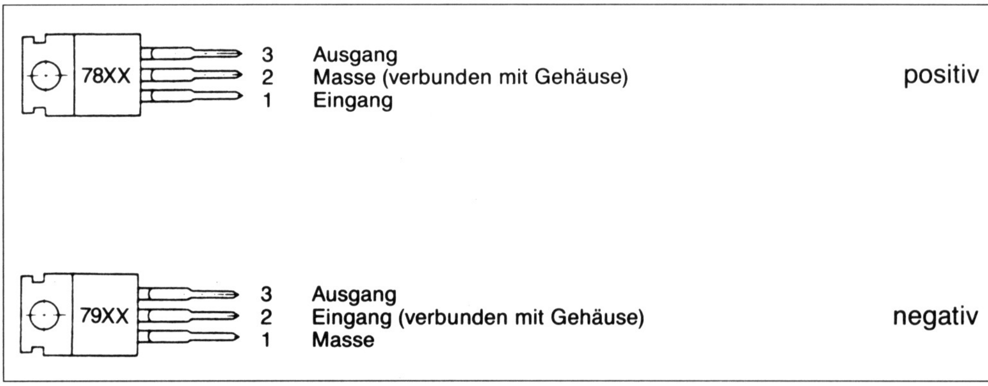
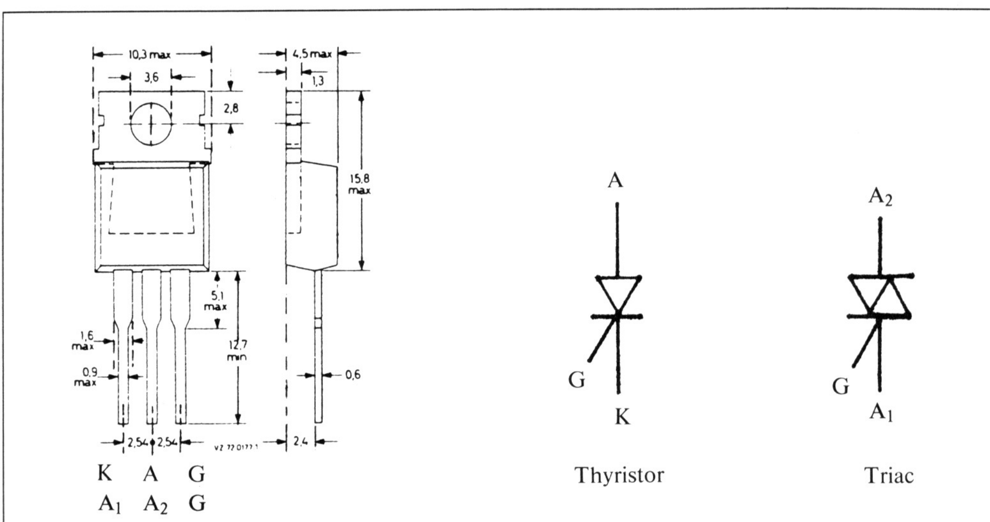


Bild A.6: Positiv- und Negativspannungsregler 78XX und 79XX



bei Thyristoren z. B. TYN 058...1008 (8 A)
bei Triacs z. B. TIC 226 D (8 A)

Bild A.7: Schaltsymbole von Thyristor Triac mit Pinbelegung im Gehäuse TO 220

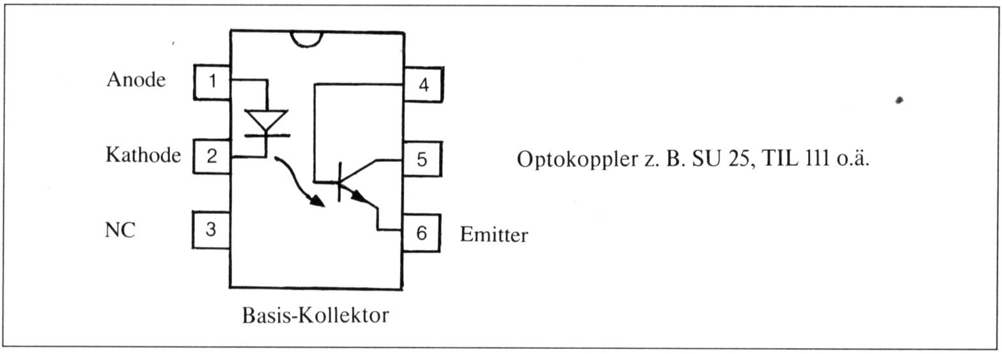


Bild A.8: Pinbelegung gängiger Optokoppler

RAMs

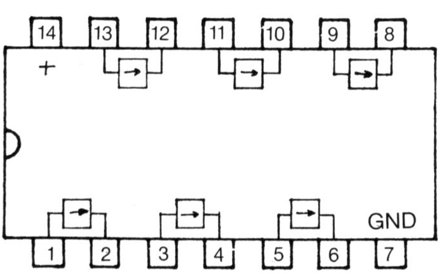
EPROMs

65256	6264	6116	27512	27256	27128	2764	2732	2716	2708	Pin (28)	Pin (24)
A_{14}	NC		A_{15}	V_{PP}	V_{PP}	V_{PP}				1	
A_{12}	A_{12}		A_{12}	A_{12}	A_{12}	A_{12}				2	
A_7	A_7	A_7	A_7	A_7	A_7	A_7	A_7	A_7	A_7	3	1
A_6	A_6	A_6	A_6	A_6	A_6	A_6	A_6	A_6	A_6	4	2
A_5	A_5	A_5	A_5	A_5	A_5	A_5	A_5	A_5	A_5	5	3
A_4	A_4	A_4	A_4	A_4	A_4	A_4	A_4	A_4	A_4	6	4
A_3	A_3	A_3	A_3	A_3	A_3	A_3	A_3	A_3	A_3	7	5
A_2	A_2	A_2	A_2	A_2	A_2	A_2	A_2	A_2	A_2	8	6
A_1	A_1	A_1	A_1	A_1	A_1	A_1	A_1	A_1	A_1	9	7
A_0	A_0	A_0	A_0	A_0	A_0	A_0	A_0	A_0	A_0	10	8
D_0	D_0	D_0	D_0	D_0	D_0	D_0	D_0	D_0	D_0	11	9
D_1	D_1	D_1	D_1	D_1	D_1	D_1	D_1	D_1	D_1	12	10
D_2	D_2	D_2	D_2	D_2	D_2	D_2	D_2	D_2	D_2	13	11
0_V	0_V	0_V	0_V	0_V	0_V	0_V	0_V	0_V	0_V	14	12

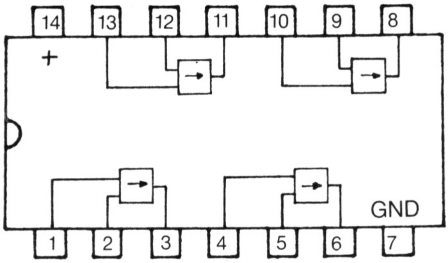
Pin (24)	Pin (28)	2708	2716	2732	2764	27128	27256	27512	6116	6264	65256
	28				+5V	+5V	+5V	+5V		+5V	+5V
	27				PGM	PGM	A_{14}	A_{14}		$\overline{WE}$	$\overline{WE}$
24	26	+5V	+5V	+5V	NC	A_{13}	A_{13}	A_{13}	+5V	CS_2	A_{13}
23	25	A_8	A_8	A_8	A_8	A_8	A_8	A_8	A_8	A_8	A_8
22	24	A_9	A_9	A_9	A_9	A_9	A_9	A_9	A_9	A_9	A_9
21	23	-5V	V_{PP}	A_{11}	A_{11}	A_{11}	A_{11}	A_{11}	$\overline{WE}$	A_{11}	A_{11}
20	22	$\overline{CS}$/WE	$\overline{OE}$	$\overline{OE}/V_{PP}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}/V_{PP}$	$\overline{OE}$	$\overline{OE}$	RFSH/$\overline{OE}$
19	21	+12V	A_{10}	A_{10}	A_{10}	A_{10}	A_{10}	A_{10}	A_{10}	A_{10}	A_{10}
18	20	V_{PP}	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CS}$	$\overline{CS_1}$	$\overline{CS}$
17	19	D_7	D_7	D_7	D_7	D_7	D_7	D_7	D_7	D_7	D_7
16	18	D_6	D_6	D_6	D_6	D_6	D_6	D_6	D_6	D_6	D_6
15	17	D_5	D'_5	D_5	D_5	D_5	D_5	D_5	D_5	D_5	D_5
14	16	D_4	D_4	D_4	D_4	D_4	D_4	D_4	D_4	D_4	D_4
13	15	D_3	D_3	D_3	D_3	D_3	D_3	D_3	D_3	D_3	D_3

Bild A.9: Die Pinbelegung der wichtigsten Speicherbausteine

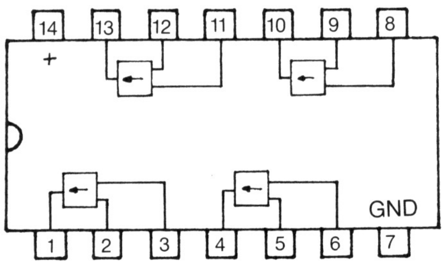
Die nachfolgenden Bilder geben Auskunft über die Pinbelegung von Logik Bausteinen:



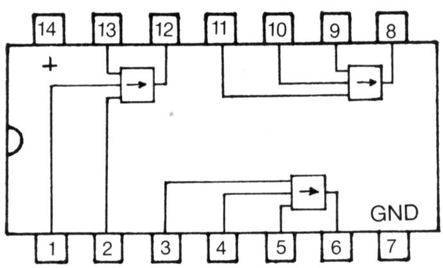
7404 Inverter
7405 Inverter, offener Kollektor
7406 Inverter, offener Kollektor, 30 V
7407 Treiber, offener Kollektor, 30 V
7416 Inverter, offener Kollektor, 15 V
7417 Treiber, offener Kollektor, 15 V
4069 Inverter, CMOS



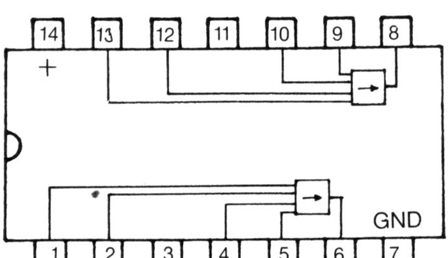
7400 NAND
7408 AND
7403 NAND, offener Kollektor
7409 AND, offener Kollektor
7426 NAND, offener Kollektor, 15 V
7409-51 AND, offener Kollektor, 15 V
7424 NAND, Schmitt-Trigger
7432 OR
7486 EXOR



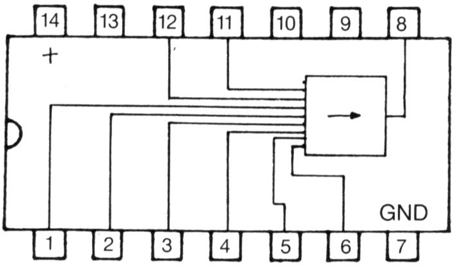
7402 NOR
7428 NOR, für hohe Ausgangslast
7433 NOR, offener Kollektor



7410 NAND
7411 AND
7412 NAND, offener Kollektor
7415 AND, offener Kollektor
7427 NOR



7420 NAND
7421 AND
7422 NAND, offener Kollektor, 5,5 V
7440 NAND, für hohe Ausgangslast
7413 NAND, Schmitt-Trigger
7418 NAND, Schmitt-Trigger



7430 NAND

Zeichentabellen

Nachfolgend finden Sie eine ASCII-Tabelle. Alle 256 möglichen 8-Bit-Kombinationen sind in ihr jeweils mit der zugehörigen Hexadezimal-, Oktal- und Dezimalzahl sowie mit dem dargestellten Zeichen im PC-Code enthalten.

Beim Amiga sind die deutschen Sonderzeichen und die entsprechenden ASCII-Zeichen gleichzeitig darstellbar. Das setzt voraus, daß verschiedene Codes benutzt werden. Diese Codes sind leider nicht genormt und daher von Gerät zu Gerät unterschiedlich. Tabelle B.1 zeigt die Kodierung beim Amiga und beim Atari ST.

Zeichen	ASCII	Amiga	Atari ST
ä	$7B	$E4	$84
ü	$7D	$FC	$81
ö	$7C	$F6	$94
Ä	$5B	$C4	$8E
Ü	$5D	$DC	$9A
Ö	$5C	$D6	$94
ß	$7E	$DF	$9E

Tabelle B.1: Die deutschen Sonderzeichen werden bei jedem Gerät anders kodiert

Die Bedeutung der Steuerzeichen (ASCII 0 bis 32 und 127) ist in Tabelle B.2 gesondert erläutert:

Dez.	Abk.	Bedeutung	Funktion
0	NUL	Null	keine Operation
1	SOH	Start of Heading	Vorspannanfang
2	STX	Start of Text	Textanfang
3	ETX	End of Text	Textende
4	EOT	End of Transmission	Übertragungsende
5	ENQ	Enquiry	Stationsanruf
6	ACK	Acknowledge	Bestätigung
7	BEL	Bell	Klingel
8	BS	Backspace	Rückwärtsschritt
9	HT	Horizontal Tabulation	Horizontaltabulator
10	LF	Line Feed	Zeilenvorschub
11	VT	Vertical Tabulation	Vertikaltabulator
12	FF	Form Feed	Formularvorschub
13	CR	Carriage Return	Wagenrücklauf
14	SO	Shift Out	Umschalten
15	SI	Shift In	Zurückschalten
16	DLE	Data Link Escape	Austritt aus der Datenverbindung
17	DC1	Device Control 1	Gerätesteuerung 1
18	DC2	Device Control 2	Gerätesteuerung 2
19	DC3	Device Control 3	Gerätesteuerung 3
20	DC4	Device Control 4	Gerätesteuerung 4
21	NAC	Negative Acknowledge	Fehlermeldung
22	SYN	Synchronous Idle	Synchronisierung
23	ETB	End of Transmitted Block	Datenblockende
24	CAN	Cancel	Ungültig
25	EM	End of Medium	Datenträgerende
26	SUB	Substitute Character	Zeichen ersetzen
27	ESC	Escape	Rücksprung
28	FS	File Separator	File-Trennung
29	GS	Group Separator	Gruppentrennung
30	RS	Record Separator	Untergruppentrennung
31	US	Unit Separator	Einheitentrennung
32	SP	Space	Leerschritt
127	DEL	Delete, Rub Out	Löschzeichen

Tabelle B.2: Die ASCII-Steuercodes

Anhang B

Zeichendarstellung Code-Tabelle

Name	Zeichen	Dez.	Okt.	Hex.	Binär	Taste/Zeichen
NUL		000_D	000_O	00_H	00000000_B	^@
SOH	☺	001_D	001_O	01_H	00000001_B	^A
STX	☻	002_D	002_O	02_H	00000010_B	^B
ETX	♥	003_D	003_O	03_H	00000011_B	^C
EOT	♦	004_D	004_O	04_H	00000100_B	^D
ENQ	♣	005_D	005_O	05_H	00000101_B	^E
ACK	♠	006_D	006_O	06_H	00000110_B	^F
BEL	•	007_D	007_O	07_H	00000111_B	^G
BS	◘	008_D	010_O	08_H	00001000_B	^H
HT	○	009_D	011_O	09_H	00001001_B	^I
LF	◙	010_D	012_O	$0A_H$	00001010_B	^J
VT	♂	011_D	013_O	$0B_H$	00001011_B	^K
FF	♀	012_D	014_O	$0C_H$	00001100_B	^L
CR	♪	013_D	015_O	$0D_H$	00001101_B	^M
SO	♫	014_D	016_O	$0E_H$	00001110_B	^N
SI	☼	015_D	017_O	$0F_H$	00001111_B	^O
DLE	►	016_D	020_O	10_H	00010000_B	^P
DC1	◄	017_D	021_O	11_H	00010001_B	^Q
DC2	↕	018_D	022_O	12_H	00010010_B	^R
DC3	‼	019_D	023_O	13_H	00010011_B	^S
DC4	¶	020_D	024_O	14_H	00010100_B	^T
NAK	§	021_D	025_O	15_H	00010101_B	^U
SYN	▬	022_D	026_O	16_H	00010110_B	^V
ETB	↨	023_D	027_O	17_H	00010111_B	^W
CAN	↑	024_D	030_O	18_H	00011000_B	^X
EM	↓	025_D	031_O	19_H	00011001_B	^Y
SUB	→	026_D	032_O	$1A_H$	00011010_B	^Z
ESC	←	027_D	033_O	$1B_H$	00011011_B	^[
FS	∟	028_D	034_O	$1C_H$	00011100_B	^\
GS	↔	029_D	035_O	$1D_H$	00011101_B	^]
RS	▲	030_D	036_O	$1E_H$	00011110_B	^^
US	▼	031_D	037_O	$1F_H$	00011111_B	^_
		032_D	040_O	20_H	00100000_B	' '
	!	033_D	041_O	21_H	00100001_B	'!'
	"	034_D	042_O	22_H	00100010_B	'"'
	#	035_D	043_O	23_H	00100011_B	'#'
	$	036_D	044_O	24_H	00100100_B	'$'
	%	037_D	045_O	25_H	00100101_B	'%'
	&	038_D	046_O	26_H	00100110_B	'&'
	'	039_D	047_O	27_H	00100111_B	'''
	(	040_D	050_O	28_H	00101000_B	'('
	)	041_D	051_O	29_H	00101001_B	')'
	*	042_D	052_O	$2A_H$	00101010_B	'*'
	+	043_D	053_O	$2B_H$	00101011_B	'+'
	,	044_D	054_O	$2C_H$	00101100_B	','
	-	045_D	055_O	$2D_H$	00101101_B	'-'
	.	046_D	056_O	$2E_H$	00101110_B	'.'
	/	047_D	057_O	$2F_H$	00101111_B	'/'
	0	048_D	060_O	30_H	00110000_B	'0'
	1	049_D	061_O	31_H	00110001_B	'1'
	2	050_D	062_O	32_H	00110010_B	'2'
	3	051_D	063_O	33_H	00110011_B	'3'
	4	052_D	064_O	34_H	00110100_B	'4'
	5	053_D	065_O	35_H	00110101_B	'5'
	6	054_D	066_O	36_H	00110110_B	'6'
	7	055_D	067_O	37_H	00110111_B	'7'
	8	056_D	070_O	38_H	00111000_B	'8'
	9	057_D	071_O	39_H	00111001_B	'9'
	:	058_D	072_O	$3A_H$	00111010_B	':'
	;	059_D	073_O	$3B_H$	00111011_B	';'
	<	060_D	074_O	$3C_H$	00111100_B	'<'
	=	061_D	075_O	$3D_H$	00111101_B	'='
	>	062_D	076_O	$3E_H$	00111110_B	'>'
	?	063_D	077_O	$3F_H$	00111111_B	'?'

Bild B.1: PC-Code-Tabelle mit Zeichendarstellung

Bild B.1: PC-Code-Tabelle mit Zeichendarstellung (Fortsetzung)

Dez.	Okt.	Hex.	Binär	Zeichen
064_D	100_O	40_H	01000000_B	'§'
065_D	101_O	41_H	01000001_B	'A'
066_D	102_O	42_H	01000010_B	'B'
067_D	103_O	43_H	01000011_B	'C'
068_D	104_O	44_H	01000100_B	'D'
069_D	105_O	45_H	01000101_B	'E'
070_D	106_O	46_H	01000110_B	'F'
071_D	107_O	47_H	01000111_B	'G'
072_D	110_O	48_H	01001000_B	'H'
073_D	111_O	49_H	01001001_B	'I'
074_D	112_O	$4A_H$	01001010_B	'J'
075_D	113_O	$4B_H$	01001011_B	'K'
076_D	114_O	$4C_H$	01001100_B	'L'
077_D	115_O	$4D_H$	01001101_B	'M'
078_D	116_O	$4E_H$	01001110_B	'N'
079_D	117_O	$4F_H$	01001111_B	'O'
080_D	120_O	50_H	01010000_B	'P'
081_D	121_O	51_H	01010001_B	'Q'
082_D	122_O	52_H	01010010_B	'R'
083_D	123_O	53_H	01010011_B	'S'
084_D	124_O	54_H	01010100_B	'T'
085_D	125_O	55_H	01010101_B	'U'
086_D	126_O	56_H	01010110_B	'V'
087_D	127_O	57_H	01010111_B	'W'
088_D	130_O	58_H	01011000_B	'X'
089_D	131_O	59_H	01011001_B	'Y'
090_D	132_O	$5A_H$	01011010_B	'Z'
091_D	133_O	$5B_H$	01011011_B	'Ä'
092_D	134_O	$5C_H$	01011100_B	'Ö'
093_D	135_O	$5D_H$	01011101_B	'Ü'
094_D	136_O	$5E_H$	01011110_B	'^'
095_D	137_O	$5F_H$	01011111_B	'_'
096_D	140_O	60_H	01100000_B	'''
097_D	141_O	61_H	01100001_B	'a'
098_D	142_O	62_H	01100010_B	'b'
099_D	143_O	63_H	01100011_B	'c'
100_D	144_O	64_H	01100100_B	'd'
101_D	145_O	65_H	01100101_B	'e'
102_D	146_O	66_H	01100110_B	'f'
103_D	147_O	67_H	01100111_B	'g'
104_D	150_O	68_H	01101000_B	'h'
105_D	151_O	69_H	01101001_B	'i'
106_D	152_O	$6A_H$	01101010_B	'j'
107_D	153_O	$6B_H$	01101011_B	'k'
108_D	154_O	$6C_H$	01101100_B	'l'
109_D	155_O	$6D_H$	01101101_B	'm'
110_D	156_O	$6E_H$	01101110_B	'n'
111_D	157_O	$6F_H$	01101111_B	'o'
112_D	160_O	70_H	01110000_B	'p'
113_D	161_O	71_H	01110001_B	'q'
114_D	162_O	72_H	01110010_B	'r'
115_D	163_O	73_H	01110011_B	's'
116_D	164_O	74_H	01110100_B	't'
117_D	165_O	75_H	01110101_B	'u'
118_D	166_O	76_H	01110110_B	'v'
119_D	167_O	77_H	01110111_B	'w'
120_D	170_O	78_H	01111000_B	'x'
121_D	171_O	79_H	01111001_B	'y'
122_D	172_O	$7A_H$	01111010_B	'z'
123_D	173_O	$7B_H$	01111011_B	'ä'
124_D	174_O	$7C_H$	01111100_B	'ö'
125_D	175_O	$7D_H$	01111101_B	'ü'
126_D	176_O	$7E_H$	01111110_B	'ß'
127_D	177_O	$7F_H$	01111111_B	' '

Fortsetzung
siehe nächste Seite

Bild B.1: PC-Code-Tabelle mit Zeichendarstellung (Fortsetzung)

Zeichen	Dez.	Okt.	Hex.	Binär	
Ç	128_D	200_O	80_H	10000000_B	^@
ü	129_D	201_O	81_H	10000001_B	^A
é	130_D	202_O	82_H	10000010_B	^B
â	131_D	203_O	83_H	10000011_B	^C
ä	132_D	204_O	84_H	10000100_B	^D
à	133_D	205_O	85_H	10000101_B	^E
å	134_D	206_O	86_H	10000110_B	^F
ç	135_D	207_O	87_H	10000111_B	^G
ê	136_D	210_O	88_H	10001000_B	^H
ë	137_D	211_O	89_H	10001001_B	^I
è	138_D	212_O	$8A_H$	10001010_B	^J
ï	139_D	213_O	$8B_H$	10001011_B	^K
î	140_D	214_O	$8C_H$	10001100_B	^L
ì	141_D	215_O	$8D_H$	10001101_B	^M
Ä	142_D	216_O	$8E_H$	10001110_B	^N
Å	143_D	217_O	$8F_H$	10001111_B	^O

Zeichen	Dez.	Okt.	Hex.	Binär	
É	144_D	220_O	90_H	10010000_B	^P
æ	145_D	221_O	91_H	10010001_B	^Q
Æ	146_D	222_O	92_H	10010010_B	^R
ô	147_D	223_O	93_H	10010011_B	^S
ö	148_D	224_O	94_H	10010100_B	^T
ò	149_D	225_O	95_H	10010101_B	^U
û	150_D	226_O	96_H	10010110_B	^V
ù	151_D	227_O	97_H	10010111_B	^W
ÿ	152_D	230_O	98_H	10011000_B	^X
Ö	153_D	231_O	99_H	10011001_B	^Y
Ü	154_D	232_O	$9A_H$	10011010_B	^Z
¢	155_D	233_O	$9B_H$	10011011_B	^[
£	156_D	234_O	$9C_H$	10011100_B	^\
¥	157_D	235_O	$9D_H$	10011101_B	^]
₧	158_D	236_O	$9E_H$	10011110_B	^^
ƒ	159_D	237_O	$9F_H$	10011111_B	^_

Zeichen	Dez.	Okt.	Hex.	Binär	
á	160_D	240_O	$A0_H$	10100000_B	' '
í	161_D	241_O	$A1_H$	10100001_B	'!'
ó	162_D	242_O	$A2_H$	10100010_B	'"'
ú	163_D	243_O	$A3_H$	10100011_B	'#'
ñ	164_D	244_O	$A4_H$	10100100_B	'$'
Ñ	165_D	245_O	$A5_H$	10100101_B	'%'
ª	166_D	246_O	$A6_H$	10100110_B	'&'
º	167_D	247_O	$A7_H$	10100111_B	'''
¿	168_D	250_O	$A8_H$	10101000_B	'('
⌐	169_D	251_O	$A9_H$	10101001_B	')'
¬	170_D	252_O	AA_H	10101010_B	'*'
½	171_D	253_O	AB_H	10101011_B	'+'
¼	172_D	254_O	AC_H	10101100_B	','
¡	173_D	255_O	AD_H	10101101_B	'-'
«	174_D	256_O	AE_H	10101110_B	'.'
»	175_D	257_O	AF_H	10101111_B	'/'

Zeichen	Dez.	Okt.	Hex.	Binär	
░	176_D	260_O	$B0_H$	10110000_B	'0'
▒	177_D	261_O	$B1_H$	10110001_B	'1'
▓	178_D	262_O	$B2_H$	10110010_B	'2'
│	179_D	263_O	$B3_H$	10110011_B	'3'
┤	180_D	264_O	$B4_H$	10110100_B	'4'
╡	181_D	265_O	$B5_H$	10110101_B	'5'
╢	182_D	266_O	$B6_H$	10110110_B	'6'
╖	183_D	267_O	$B7_H$	10110111_B	'7'
╕	184_D	270_O	$B8_H$	10111000_B	'8'
╣	185_D	271_O	$B9_H$	10111001_B	'9'
║	186_D	272_O	BA_H	10111010_B	':'
╗	187_D	273_O	BB_H	10111011_B	';'
╝	188_D	274_O	BC_H	10111100_B	'<'
╜	189_D	275_O	BD_H	10111101_B	'='
╛	190_D	276_O	BE_H	10111110_B	'>'
┐	191_D	277_O	BF_H	10111111_B	'?'

Fortsetzung siehe nächste Seite

Bild B.1: PC-Code-Tabelle mit Zeichendarstellung (Fortsetzung)

Zeichen	D	O	H	B	Zeichen
└	192	300	C0	11000000	'§'
┴	193	301	C1	11000001	'A'
┬	194	302	C2	11000010	'B'
├	195	303	C3	11000011	'C'
─	196	304	C4	11000100	'D'
┼	197	305	C5	11000101	'E'
╞	198	306	C6	11000110	'F'
╟	199	307	C7	11000111	'G'
╚	200	310	C8	11001000	'H'
╔	201	311	C9	11001001	'I'
╩	202	312	CA	11001010	'J'
╦	203	313	CB	11001011	'K'
╠	204	314	CC	11001100	'L'
═	205	315	CD	11001101	'M'
╬	206	316	CE	11001110	'N'
╧	207	317	CF	11001111	'O'
╨	208	320	D0	11010000	'P'
╤	209	321	D1	11010001	'Q'
╥	210	322	D2	11010010	'R'
╙	211	323	D3	11010011	'S'
╘	212	324	D4	11010100	'T'
╒	213	325	D5	11010101	'U'
╓	214	326	D6	11010110	'V'
╫	215	327	D7	11010111	'W'
╪	216	330	D8	11011000	'X'
┘	217	331	D9	11011001	'Y'
┌	218	332	DA	11011010	'Z'
█	219	333	DB	11011011	'Ä'
▄	220	334	DC	11011100	'Ö'
▌	221	335	DD	11011101	'Ü'
▐	222	336	DE	11011110	'^'
▀	223	337	DF	11011111	'_'
α	224	340	E0	11100000	'`'
ß	225	341	E1	11100001	'a'
Γ	226	342	E2	11100010	'b'
π	227	343	E3	11100011	'c'
Σ	228	344	E4	11100100	'd'
σ	229	345	E5	11100101	'e'
µ	230	346	E6	11100110	'f'
τ	231	347	E7	11100111	'g'
Φ	232	350	E8	11101000	'h'
Θ	233	351	E9	11101001	'i'
Ω	234	352	EA	11101010	'j'
δ	235	353	EB	11101011	'k'
∞	236	354	EC	11101100	'l'
φ	237	355	ED	11101101	'm'
ε	238	356	EE	11101110	'n'
∩	239	357	EF	11101111	'o'
≡	240	360	F0	11110000	'p'
±	241	361	F1	11110001	'q'
≥	242	362	F2	11110010	'r'
≤	243	363	F3	11110011	's'
⌠	244	364	F4	11110100	't'
⌡	245	365	F5	11110101	'u'
÷	246	366	F6	11110110	'v'
≈	247	367	F7	11110111	'w'
°	248	370	F8	11111000	'x'
∙	249	371	F9	11111001	'y'
·	250	372	FA	11111010	'z'
√	251	373	FB	11111011	'ä'
ⁿ	252	374	FC	11111100	'ö'
²	253	375	FD	11111101	'ü'
■	254	376	FE	11111110	'ß'
	255	377	FF	11111111	' '

Anhang C

C.1 Empfehlenswerte Literatur

Amiga:

R. Peck, S. Deyl, J. Miner: Amiga Hardware Manual, Los Gatos 1985
Commodore Amiga: A500/A2000 Technical Reference Manual, 1986/1987
Commodore-Amiga Inc.: Amiga DOS-Handbuch, Haar 1987

Datenbücher:

IWT Verlag: TTL-Taschenbuch Teil 1, München 1983
IWT Verlag: TTL-Taschenbuch Teil 2, München 1983
IWT Verlag: CMOS-Taschenbuch Band 1, München 1981
VALVO-Bauelemente: SC 68000–System, Hamburg 1986

Schaltungstechnik:

Beckmann, Ernst: Experimente Elektronik, Köln 1977
Bernstein, Herbert: Elektronik Hobby, München 1982
Gerlach, Uwe: Hardwarebasteleien zum C64/C128, Haar 1987

Videotechnik:

Zastrow, Peter: Fernsehempfangstechnik, Frankfurt 1977
Limmann, Pelka: Fernsehtechnik ohne Ballast, München 1981

C.2 Bezugsadressen für elektronische Bauelemente

Die benötigten Bauelemente können Sie über den Fachhandel beziehen, am besten bei einem Elektronikgeschäft in Ihrer Nähe. Dort kann man Ihnen sicherlich auch bei Fragen weiterhelfen. Kommen größere Mengen zusammen, dann lohnt sich eine Bestellung beim Versandhandel. Hier einige der größten Anbieter für Hobbybedarf:

Bürklin Elektronik, Schillerstraße 40, 8000 München 2

Conrad Electronic GmbH, Klaus-Conrad-Straße 1, 8452 Hirschau

Völkner Electronic, Marienberger Straße 10, 3300 Braunschweig

G. Simons electronic, Meisenweg 4, 5012 Bedburg

HW Elektronik, Chaussee 79, 2000 Hamburg 19 Eimsb.

Komplette Bausätze können Sie bei folgender Firma bestellen:

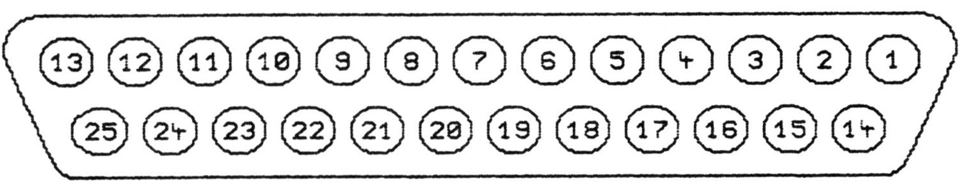
Schmotz Elektronik, Postfach 1412,
8202 Bad Aibling.

Anhang D

Die Belegung der wichtigsten Amiga-Steckverbindungen

D.1 Paralleler Port

Amiga 500/2000: weiblicher Stecker im Gerät

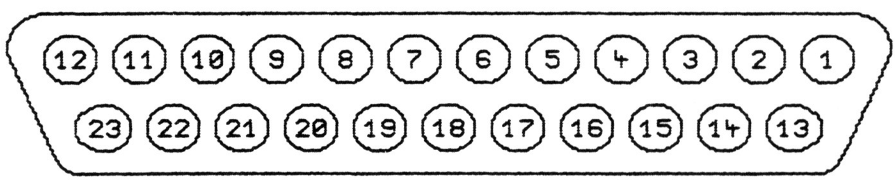


Pin	Signal	Funktion	intern		Richtg.
1	$\overline{\text{STROBE}}$	Strobe-Signal	8520-B	$\overline{\text{PC}}$	aus
2	D0	Datenbit 0	8520-B	PB0	ein/aus
3	D1	Datenbit 1	8520-B	PB1	ein/aus
4	D2	Datenbit 2	8520-B	PB2	ein/aus
5	D3	Datenbit 3	8520-B	PB3	ein/aus
6	D4	Datenbit 4	8520-B	PB4	ein/aus
7	D5	Datenbit 5	8520-B	PB5	ein/aus
8	D6	Datenbit 6	8520-B	PB6	ein/aus
9	D7	Datenbit 7	8520-B	PB7	ein/aus
10	$\overline{\text{ACK}}$	Acknowledge	8520-B	$\overline{\text{FLAG}}$	ein
11	BUSY	Drucker aktiv	8520-A	PA0	ein/aus
			8520-A	SP	ein/aus

Pin	Signal	Funktion	intern	Richtg.
12	POUT	Papierende	8520-A PA1	ein/aus
			8520-A CNT	ein/aus
13	SEL	Drucker on-line	8520-A PA2	ein/aus
14	+5V	Versorgungsspannung	+5V (47 Ohm)	
15	---	(nicht belegt)		
16	$\overline{\text{RESET}}$	Reset-Leitung	gepuffert	aus
17	GND	Signalmasse	GND	
18	GND	Signalmasse	GND	
19	GND	Signalmasse	GND	
20	GND	Signalmasse	GND	
21	GND	Signalmasse	GND	
22	GND	Signalmasse	GND	
23	GND	Signalmasse	GND	
24	GND	Signalmasse	GND	
25	GND	Signalmasse	GND	

D.2 Disk-Stecker

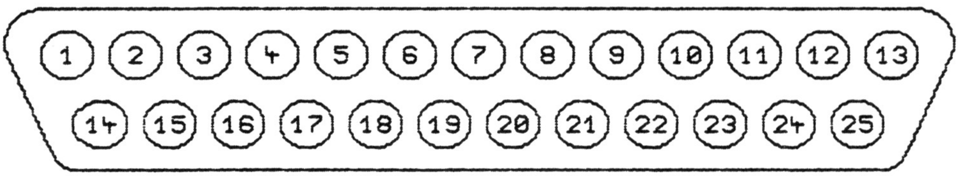
Amiga 500/1000/2000: weiblicher Stecker im Gerät



Pin	Signal	Anschluß intern	Funktion
1	$\overline{\text{RDY}}$	8520-A PA5	Disk Ready
2	$\overline{\text{DKRD}}$	PAULA	Disk Read Data
3	GND	GND	Signalmasse
4	GND	GND	Signalmasse
5	GND	GND	Signalmasse
6	GND	GND	Signalmasse
7	GND	GND	Signalmasse

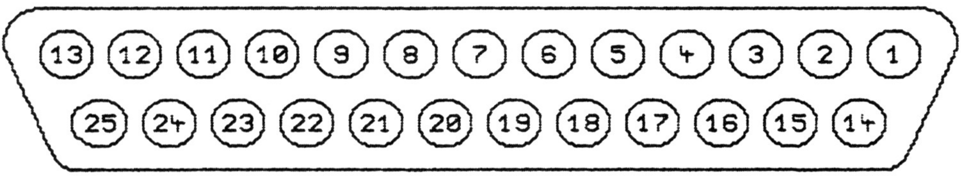
Pin	Signal	Anschluß intern	Funktion
8	$\overline{\text{MTRXD}}$	8520-B PB7	Disk-Motor an (wird vom Laufwerk bei Aktivierung des entsprechenden Select-Signals zwischengespeichert)
9	$\overline{\text{SEL2B}}$	8520-B PB5	Disk-Select Laufwerk 2 (bei Amiga 2000 $\overline{\text{SEL3B}}$)
10	$\overline{\text{DRESB}}$	Reset	Systemreset gepuffert
11	$\overline{\text{CHNGE}}$	8520-A PA2	HIGH = Disk entnommen (wird bis zum nächsten Step-Impuls gelatcht)
12	+5V	+5 Volt	Versorgungsspannung
13	$\overline{\text{SIDEB}}$	8520-B PB2	Disk-Kopfauswahl: 0 oben 1 unten
14	$\overline{\text{WPRO}}$	8520-A PA3	Disk ist schreibgeschützt
15	$\overline{\text{TK0}}$	8520-A PA4	Disk-Kopf auf Spur 0
16	$\overline{\text{DKWEB}}$	PAULA	Disk Write Enable
17	$\overline{\text{DKWDB}}$	PAULA	Disk Write Data
18	$\overline{\text{STEPB}}$	8520-B PB0	Stepperschritte für Kopfbewegung. Muß nach jedem Puls unbedingt wieder HIGH werden
19	$\overline{\text{DIRB}}$	8520-B PB1	Step-Richtung für Kopfbewegung. 0 zur Diskmitte 1 nach Außen (Spur 0 ist außen)
20	$\overline{\text{SEL3B}}$	8520-B PB6	Disk-Select Laufwerk 3
21	$\overline{\text{SEL1B}}$	8520-B PB4	Disk-Select Laufwerk 1 (bei Amiga 2000 -SEL2B)
22	$\overline{\text{INDEX}}$	8520-B $\overline{\text{FLAG}}$	Disk-Index-Signal
23	+12V	+12 Volt	Versorgungsspannung

Amiga 1000: männlicher Stecker im Gerät



Pin	Signal	Funktion	intern	Richtg.
1	$\overline{\text{STROBE}}$	Strobe-Signal	8520-B $\overline{\text{PC}}$	aus
2	D0	Datenbit 0	8520-B PB0	ein/aus
3	D1	Datenbit 1	8520-B PB1	ein/aus
4	D2	Datenbit 2	8520-B PB2	ein/aus
5	D3	Datenbit 3	8520-B PB3	ein/aus
6	D4	Datenbit 4	8520-B PB4	ein/aus
7	D5	Datenbit 5	8520-B PB5	ein/aus
8	D6	Datenbit 6	8520-B PB6	ein/aus
9	D7	Datenbit 7	8520-B PB7	ein/aus
10	$\overline{\text{ACK}}$	Acknowledge	8520-B $\overline{\text{FLAG}}$	ein
11	BUSY	Drucker aktiv	8520-A PA0	ein/aus
			8520-A SP	ein/aus
12	POUT	Papierende	8520-A PA1	ein/aus
			8520-A CNT	ein/aus
13	SEL	Drucker on-line	8520-A PA2	ein/aus
14	GND	Signalmasse	GND I	
15	GND	Signalmasse	GND I	
16	GND	Signalmasse	GND I	
17	GND	Signalmasse	GND I	
18	GND	Signalmasse	GND I	
19	GND	Signalmasse	GND I	
20	GND	Signalmasse	GND I	
21	GND	Signalmasse	GND I	
22	GND	Signalmasse	GND I	
23	+5V	Versorgungsspannung	+5V I	
24	---	(nicht belegt)		
25	$\overline{\text{RESET}}$	Reset-Leitung	gepuffert	aus

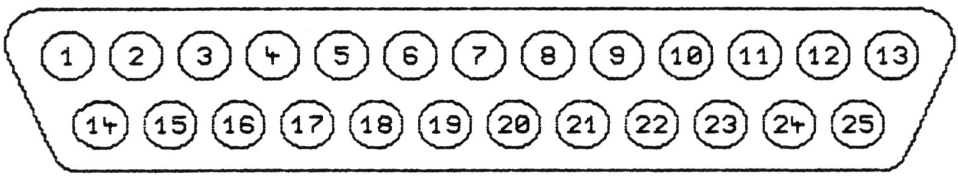
Amiga 1000: weiblicher Stecker im Gerät



Pin	Signal	Funktion	intern	Richtg.
1	SHIELD	Schirmung	GND	
2	TXD	Sendedaten	PAULA	aus
3	RXD	Empfangsdaten	PAULA	ein
4	RTS	Sendebereitschaft	8520-A PA6	aus
5	CTS	Sendefreigabe	8520-A PA4	ein
6	DSR	Empfänger bereit	8520-A PA3	ein
7	GND	Signalmasse	GND	
8	CD	Trägererkennung	8520-A PA5	ein
9	---	(nicht belegt)		
10	---	(nicht belegt)		
11	---	(nicht belegt)		
12	---	(nicht belegt)		
13	---	(nicht belegt)		
14	–5V	Versorgungsspannung	–5V	
15	AUDO	Tonausgang	linker Kanal	aus
16	AUDI	Toneingang	wird zum rechten Kanal gemischt	
17	EB	Port-Takt	GND	
18	$\overline{\text{INT2}}$	Interrupt Level 2	PAULA	ein
19	---	(nicht belegt)		
20	DTR	Endgerät bereit	8520-A PA7	aus
21	+5V	Versorgungsspannung	+5V	
22	---	(nicht belegt)		
23	+12V	Versorgungsspannung	+12V	
24	$\overline{\text{C2}}$	3,58 MHz-Takt	1/2 Proz.takt	aus
25	$\overline{\text{RESB}}$	System-Reset	gepuffert	aus

D.3 RS-232-Port

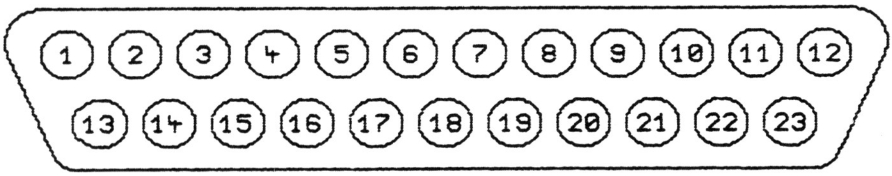
Amiga 500/2000: männlicher Stecker im Gerät



Pin	Signal	Funktion	intern	Richtg.
1	SHIELD	Schirmung	GND	
2	TXD	Sendedaten	PAULA	aus
3	RXD	Empfangsdaten	PAULA	ein
4	RTS	Sendebereitschaft	8520-A PA6	aus
5	CTS	Sendefreigabe	8520-A PA4	ein
6	DSR	Empfänger bereit	8520-A PA3	ein
7	GND	Signalmasse	GND	
8	CD	Trägererkennung	8520-A PA5	ein
9	+12V	Versorgungsspannung	+12V (470 Ohm)	
10	-12V	Versorgungsspannung	-12V (470 Ohm)	
11	AUDO	Tonausgang	linker Kanal	aus
12	---	(nicht belegt)		
13	---	(nicht belegt)		
14	---	(nicht belegt)		
15	---	(nicht belegt)		
16	---	(nicht belegt)		
17	---	(nicht belegt)		
18	AUDI	Toneingang	wird zum rechten Kanal gemischt	
19	---	(nicht belegt)		
20	DTR	Endgerät bereit	8520-A PA7	aus
21	---	(nicht belegt)		
22	RI	Rufsignalerkennung		ein
23	---	(nicht belegt)		
24	---	(nicht belegt)		
25	---	(nicht belegt)		

D.4 Paralleler Port

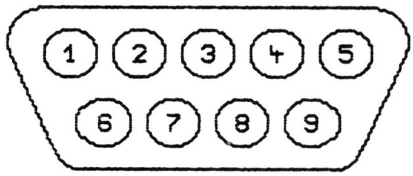
Amiga 500/1000/2000: männlicher Stecker im Gerät



Pin	Signal	Funktion	intern	Richtg.
1	$\overline{XCLK}$	Extern Clock	Taktversorgung	aus
2	$\overline{XCLKEN}$	Extern Clock Enable	Taktversorgung	ein/aus
3	R	analog Rot	Video-Wandler	aus
4	G	analog Grün	Video-Wandler	aus
5	B	analog Blau	Video-Wandler	aus
6	DI	digitale Intensität	DENISE	aus
7	DB	digital Blau	DENISE	aus
8	DG	digital Grün	DENISE	aus
9	DR	digital Rot	DENISE	aus
10	$\overline{CSY}$	Composite-Sync	AGNUS	ein/aus
11	$\overline{HSY}$	Horizontal-Sync	AGNUS	ein/aus
12	$\overline{VSY}$	Vertikal-Sync	AGNUS	ein/aus
13	GND	Masse für $\overline{XCLK}$	Digital-Masse	
14	$\overline{ZD}$	Hintergrundfarbe	DENISE	aus
15	$\overline{C1}$	Taktsignal	Taktversorgung	aus
16	GND	Masse	Video-Masse	
17	GND	Masse	Video-Masse	
18	GND	Masse	Video-Masse	
19	GND	Masse	Video-Masse	
20	GND	Masse	Video-Masse	
21	– 5V	– 5 Volt	Netzteil	
22	+ 12V	+ 12 Volt	Netzteil	
23	+ 5V	+ 5 Volt	Netzteil	

D.5 Control-Ports

Amiga 500/1000/2000: männlicher Stecker im Gerät



Pin	Maus	Joystick/Paddle	Lightpen
1	MOUSE V	FORWARD	–
2	MOUSE H	BACK	–
3	MOUSE VQ	LEFT	–
4	MOUSE HQ	RIGHT	–
5	BUTTON 2	POT X	LP PRESS
6	BUTTON 1	FIRE	LIGHT PEN
7	+5V	+5V	+5V
8	GND	GND	GND
9	BUTTON 3	POT Y	–

Anhang E

Guru-Fehlermeldungen

Guru-Meditation Fehlercodes

Die sechzehn Ziffern eines Guru-Fehlercodes lassen sich nach folgendem Schema unterteilen:

AABBCCCC.EEEEEEEE

Die ersten beiden Ziffern (A) kennzeichnen das Subsystem, die nächsten beiden (B) den grundsätzlichen Fehler und die Vierergruppe (C) den spezifizierten Fehler. Jede Subsystem-Kennzeichnung (A) hat jeweils anders spezifizierte Vierergruppen (C). Die letzten acht Ziffern bestimmen schließlich die Adresse (Hexadezimal) des Tasks, der den Absturz verursacht hat.

Mit der Anzeige des Fehlercodes fordert der Computer den Benutzer auf, die linke Maustaste zu drücken. Mit der rechten Maustaste wird in bestimmten Fällen das neuerliche Booten umgangen. Ist die erste Ziffer kleiner oder gleich 7, springt der Computer mit dem Druck auf die rechte Maustaste direkt ins Programm zurück. Ist die erste Ziffer größer 7, ist eine Fortführung des Programms nicht mehr möglich.

Prozessor-Fehler

Der Fehlercode 00 in der ersten Zahlengruppe (Kennung A) hat eine Ausnahmestellung inne. Er steht für einen CPU-Fehler. Diese Meldungen sind durch den 68000-Prozessor und nicht durch die Amiga-Systemsoftware definiert. Dem Prozessor stehen für die Fehlerbehandlung 256 Vektoren zur Verfügung. Die ersten 64 dieser Vektoren sind durch den Prozessor definiert. Die oberen 192 Vektoren können durch den Benutzer definiert werden.

00 00 0002 Bus-Error
Timingfehler auf Adreß- oder Datenbus

00 00 0003 Address Error
Adressierungsfehler

00 00 0004 Illegal Instruction
Unzulässige Instruction

00 00 0005 Devide by Zero
Division durch Null

00 00 0006 CHK Instruction
CHK testet Register gegen die Grenzen des Zahlenbereichs. Der Fehler tritt auf,

wenn die zulässigen Grenzen nicht eingehalten werden.

00 00 0007 TRAPV Instruction
TRAPV verzweigt auf Trap-Vektor, wenn V-Flag gesetzt.

00 00 0008 Privilege Violation
Privilegverletzung

00 00 0009 Trace
Einzelschritt-Modus

00 00 000A OP Code 1010
Unbenutzter Op-Code

00 00 000B OP Code 1111
Unbenutzter Op-Code

Amiga-Systemfehlermeldungen

Alle anderen Guru-Meditations bezeichnen mit den ersten beiden Stellen (A) den Betriebssystemteil, in dem der Fehler aufgetreten ist.

Die Libraries haben in der ersten Position die Null:

01 BB CCCC Exec
02 BB CCCC Graphics
03 BB CCCC Layers
04 BB CCCC Intuition
05 BB CCCC Math
06 BB CCCC Clist
07 BB CCCC DOS
08 BB CCCC RAM
09 BB CCCC Icon
0A BB CCCC Expansion

Die Devices sind durch die Eins gekennzeichnet:

10 BB CCCC Audio
11 BB CCCC Console
12 BB CCCC GamePort
13 BB CCCC Keyboard
14 BB CCCC TrackDisk
15 BB CCCC Timer

Die Zwei definiert Resource

20 BB CCCC CIA
21 BB CCCC Disk
22 BB CCCC Misc

Die »Sonstigen« haben die Drei in der ersten Ziffer:

30 BB CCCC Bootstrap
31 BB CCCC Workbench
32 BB CCCC DiskCopy

Übergeordnete Fehler

AA 00 CCCC
Fehler nicht zuzuordnen

AA 01 CCCC Insufficient Memory
Speicherplatzmangel

AA 02 CCCC MakeLibrary Error
Library kann nicht erzeugt werden

AA 03 CCCC OpenLibrary Error
Library kann nicht geöffnet werden

AA 04 CCCC OpenDevice Error
Device kann nicht geöffnet werden

AA 05 CCCC OpenResource Error
Nichtreagieren eines Hardware-Bausteins

AA 06 CCCC I/O Error
Ein/Ausgabefehler

AA 07 CCCC No Signal
Signal fehlt

Exec Library-Codes

81 00 0001
68000 Exception Vector Checksum
Prüfsummenfehler bei Ausnahmebehandlung des Prozessors

81 00 0002
ExecBase Checksum
Prüfsummenfehler der Startadresse des Exec

81 00 0003
Library Checksum Error
Prüfsummenfehler bei der Library

81 00 0004
No Memory to Make Library
Mangelnder Speicherplatz für Library

81 00 0005
Corrupted Memory List
Zerstörte Speicherverwaltungsliste

81 00 0006
No Memory For Interrupt Servers
Mangelnder Speicherplatz für Interruptbehandlung

81 00 0007
initAPTR
Zeigerfehler

81 00 0008
Semaphore Corrupt
Semaphore zerstört

81 00 0009
Free Twice
Speicherplatz zweimal freigegeben

81 00 000A
Bogus Exception
Es wurden reservierte Vektoren verwendet

Graphics Library-Codes:

82 01 0001
No Memory for Copper Display List
Mangelnder Speicherplatz für die Copperliste

82 01 0002
No Memory for Copper Instruction List
Mangelnder Speicherplatz für die Copper Instruction-Liste

82 00 0003
Copper List Overload
Copperliste ist voll

82 00 0004
Copper Intermediate List Overload
Struktur der Copperliste zerstört

82 01 0005
No Memory for Copper List Head
Mangelnder Speicherplatz für den Kopf der Copperliste

82 01 0006
Long Frame, No Memory
Mangelnder Speicherplatz für die Copper-Liste I bei Interlace.

82 01 0007
Short Frame, No Memory
Mangelnder Speicherplatz für die Copper-Liste I bei Interlace.

82 01 0008
No Memory for Flood Fill
Mangelnder Speicherplatz zum Ausführen des Fill-Befehls

82 01 0009
Text, No Memory for TmpRas
Mangelnder Speicherplatz zum Anlegen der temporären Datei TmpRas (Temporary raster work area)

82 01 000A
No Memory for BltBitMap
Mangelnder Speicherplatz für die Blitter-Bitmap

82 01 000B
Region Memory
Speicherbereich falsch angegeben

82 01 0030
MakeVPort
Fehler beim Einrichten des ViewPort

82 01 1234
GfxNoLCM
Zwischenspeicherbereich nicht frei

Layers Library-Codes:

83 01 0001
LayersNoMem
Mangelnder Speicherplatz für die Layers

Intuition Library-Codes:

84 00 0001
Unknown Gadget-Type
Unbekannter Gadget Typ

04 00 0001
Wie oben, doch abfangbar

84 01 0002
No Memory to create Port
Mangelnder Speicherplatz für neuen Port

84 01 0003
Item Plane Alloc, No Memory
Mangelnder Speicherplatz für Darstellung der Menüleiste

84 01 0004
Sub Alloc, No Memory
Mangelnder Speicherplatz für Darstellung der Untermenüs

84 01 0005
Plane Alloc, No Memory
Mangelnder Speicherplatz für die Kopfzeile des Menüs

84 00 0006
Item Box Top Less Than Real Zero
Die obere Grenze einer Item-Box liegt unter der absoluten Null-Position

84 01 0007
No Memory To Open Screen
Mangelnder Speicherplatz zum Öffnen eines Screens

84 01 0008
Open Screen, Raster Alloc, No Memory
Mangelnder Speicherplatz für RastPort

84 00 0009
Open Sys Screen, Unknown Type
Unbekannter Screen-Typ

84 01 000A
Add SW Gadgets, No memory
Mangelnder Speicherplatz für Gadget

84 01 000B
No Memory to Open Window
Mangelnder Speicherplatz zum Öffnen eines Windows

84 00 000C
Bad State Return Entering Intuition
Fehlerhafte Statusangabe beim Öffnen von Intuition

84 00 000D
Bad Message Received by IDCMP
Fehlermeldung der »Intuition Direct Communication Message Ports«

84 00 000E
Wierd Echo Causing Incomprehension
Mangelnder Speicherplatz für Zugriff auf die »Distant Echo List«

84 00 000F
Could not Open The Console Device
Fehler beim Öffnen des Console Device

DOS Library-Codes:

07 01 0001
No Memory At Startup
Mangelnder Speicherplatz bei Startup

07 00 0002
EndTask didn't
EndTask hat fehlerhaft oder nicht gewirkt

07 00 0003
Qptk Failure
Fehler beim Übertragen eines Datenpaketes

07 00 0004
Unexpected Packet Received
Empfang eines Datenpaketes, das nicht erwartet wurde

07 00 0005
Freevec Failed
Freevec hat fehlerhaft oder nicht gewirkt

07 00 0006
Disk Block Sequence Error
Fehler bei einer Disk-Block-Sequenz

07 00 0007
Bitmap Corrupt
Fehlerhafte oder zerstörte Bitmap

07 00 0008
Key already Free
File-Nummer (Key-Nummer) bereits gelöscht

07 00 0009
Invalid Checksum
Unzulässige Prüfsumme

07 00 000A
Disk Error

Diskettenfehler

07 00 000B
Key Out Of Range
File-Nummer (Key-Nummer) außerhalb des zulässigen Bereiches

07 00 000C
Bad Overlay
Overlay Hunk nicht in Ordnung

RAM Library-Codes:

08 00 0001
Bad Segment List
Fehlerhafte Speicherverwaltungsliste

Expansion Library-Codes:

0A 00 0001
Bad Expansion Free
Hard- oder Softwarefehler bei einer Erweiterung

TrackDisk Device-Codes:

14 00 0001
Calibrate: Seek Error
Fehler beim Suchen auf Diskette

14 00 0002
Delay: Error On Timer Wait
Fehler beim Warten auf
einen Timer-Impuls

Timer Device-Codes:

15 00 0001
Bad Request
Fehler beim Zugriffsversuch

15 00 0002
Bad Supply
Fehlsteuerung durch
Netzfrequenz (Frequenz instabil)

Disk Resource-Codes:

21 00 0001
Get Unit: Already has Disk
Fehlerhaftes DiskChange-Signal

21 00 0002
Interrupt: No Active Unit
Kein aktives Laufwerk vorhanden

BootStrap-Codes

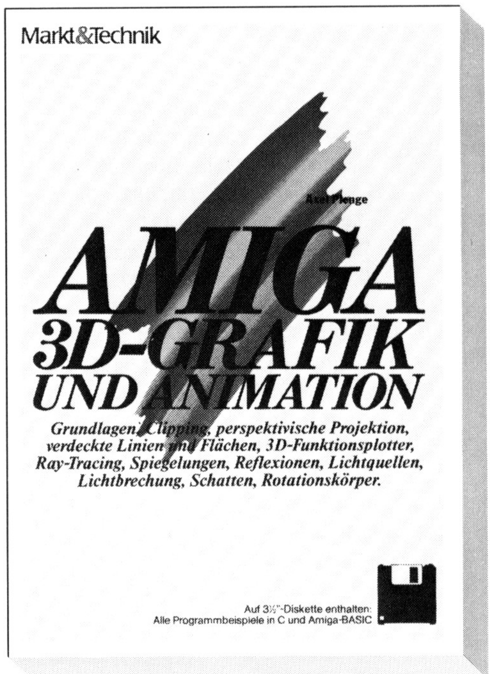
30 00 0001
Boot Code Returned an Error
DOS-Library nicht gefunden

Stichwortverzeichnis

Bücher zum Amiga



H. Knappe
Fraktale Grafik auf dem Amiga
Das Thema dieses Buches wird den meisten als ungewöhnlich erscheinen, denn es führt in die Grenzbereiche des heutigen Wissens der Mathematik und Technik.
Grundlegende Kenntnisse der Programmiersprache C und ihrer Anwendung auf dem Amiga werden vorausgesetzt. Für die nachträgliche Veränderung berechneter Bilder ist es sinnvoll, ein Malprogramm (z. B. Deluxe Paint) zu besitzen.
1988, 278 Seiten,
inkl. Diskette
Bestell-Nr. 90600
ISBN 3-89090-600-1
DM 79,–
(sFr 72,70/öS 616,-)



A. Plenge
Amiga 3-D-Grafik und Animation
Angefangen bei den einfachsten Problemstellungen lernen Sie, professionelle 3-D-Grafiken auf Ihrem Commodore Amiga zu planen, zu programmieren und darzustellen.
1988, 376 Seiten,
inkl. Diskette
Bestell-Nr. 90526
ISBN 3-89090-526-9
DM 69,–
(sFr 63,50/öS 538,-)



H. R. Henning
Grafik mit Amiga-Basic
Dieses Buch ist speziell der Grafik-Programmierung auf dem Amiga gewidmet. Der erste Teil stellt für den Anfänger alle bekannten Grafik-Befehle des Amiga-Basic vor. Mit Beginn des zweiten Teiles werden die Routinen des Betriebssystems zur Grafik-Programmierung herangezogen. Damit werden die Möglichkeiten des Basic um ein Vielfaches erweitert, und es sind Geschwindigkeiten möglich, die kaum vermuten lassen, daß dabei ein Basic-Programm abläuft.
1989, ca. 300 Seiten,
inkl. Diskette
Bestell-Nr. 90669
ISBN 3-89090-669-9
DM 59,–
(sFr 54,30/öS 460,-)

Markt&Technik
Zeitschriften · Bücher
Software · Schulung

Markt&Technik-Produkte erhalten Sie bei Ihrem Buchhändler, in Computer-Fachgeschäften oder in den Fachabteilungen der Warenhäuser.

429/901

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bitte schneiden Sie diesen Coupon aus, und schicken Sie ihn in einem Kuvert an:
Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar

Computerliteratur und Software vom Spezialisten

Vom Einsteigerbuch für den Heim- oder Personalcomputer-Neuling über professionelle Programmierhandbücher bis hin zum Elektronikbuch bieten wir Ihnen interessante und topaktuelle Titel für

• Apple-Computer • Atari-Computer • Commodore 64/128/16/116/Plus 4 • Schneider-Computer • IBM-PC, XT und Kompatible
sowie zu den Fachbereichen Programmiersprachen • Betriebssysteme (CP/M, MS-DOS, Unix, Z80) • Textverarbeitung • Datenbanksysteme • Tabellenkalkulation • Integrierte Software • Mikroprozessoren • Schulungen.
Außerdem finden Sie professionelle Spitzen-Programme in unserem preiswerten Software-Angebot für Amiga, Atari ST, Commodore 128, 128 D, 64, 16, für Schneider-Computer und für IBM-PCs und Kompatible!
Fordern Sie mit dem nebenstehenden Coupon unser neuestes Gesamtverzeichnis und unsere Programmservice-Übersichten an, mit hilfreichen Utilities, professionellen Anwendungen oder packenden Computerspielen!

Markt&Technik
Zeitschriften · Bücher
Software · Schulung

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2,
8013 Haar bei München, Telefon (089) 4613-0

709005

Adresse:

Name

Straße

Ort

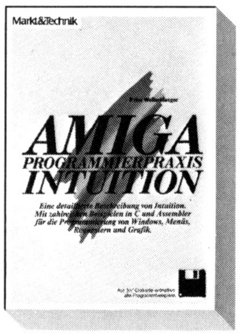
Bitte schicken Sie mir:
☐ Ihr neuestes Gesamtverzeichnis
☐ Eine Übersicht Ihres Programmservice-Angebotes aus der Zeitschrift

☐ Außerdem interessiere ich mich für folgende/n Computer:

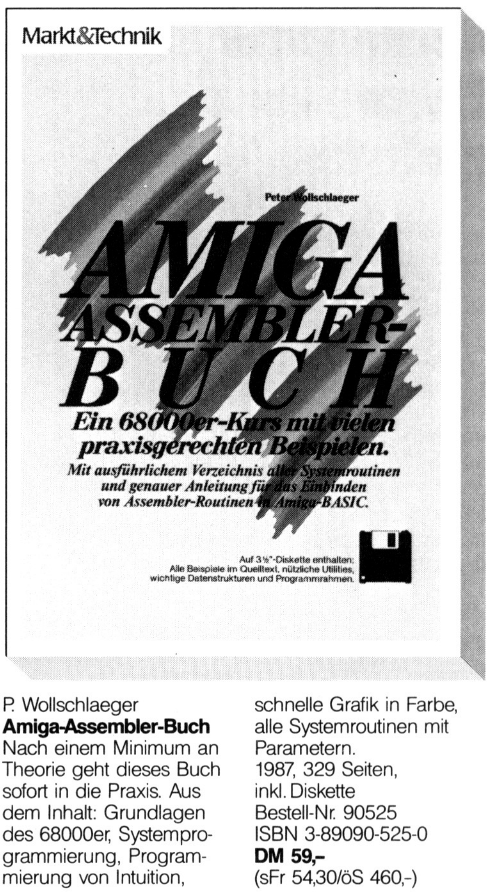
(PS: Wir speichern Ihre Daten und verpflichten uns zur Einhaltung des Bundesdatenschutzgesetzes)

Markt&Technik Verlag AG
– Unternehmensbereich Buchverlag –
Hans-Pinsel-Straße 2
D-8013 Haar bei München

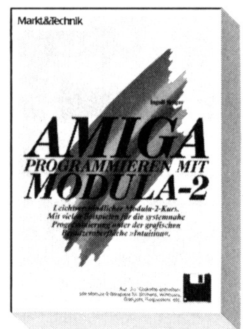
Bücher zum Amiga



P. Wollschlaeger
Amiga: Programmierpraxis Intuition
Eine detaillierte Beschreibung von Intuition! Neben der Programmierung von Fenstern, Menüs und Grafiken behandelt der Autor auch wichtige Randgebiete, wie die Ein- und Ausgabe von Texten oder Zugriff auf die Diskette.
Sie erfahren, wie ein Programm zu gestalten ist, damit es sowohl unter CLI als auch unter Intuition läuft und Multitasking-fähig ist. Mit allen Beispielen für die Programmierung von Windows, Menüs, Requestern und Grafik auf Diskette.
1988, 330 Seiten,
inkl. Diskette
Bestell-Nr. 90593
ISBN 3-89090-593-5
DM 69,–
(sFr 63,50/öS 538,-)



P. Wollschlaeger
Amiga-Assembler-Buch
Nach einem Minimum an Theorie geht dieses Buch sofort in die Praxis. Aus dem Inhalt: Grundlagen des 68000er, Systemprogrammierung, Programmierung von Intuition, schnelle Grafik in Farbe, alle Systemroutinen mit Parametern.
1987, 329 Seiten,
inkl. Diskette
Bestell-Nr. 90525
ISBN 3-89090-525-0
DM 59,–
(sFr 54,30/öS 460,-)



I. Krüger
Amiga: Programmieren mit Modula 2
Leichtverständlicher Modula-2-Kurs! Mit vielen Beispielen für die systemnahe Programmierung unter der grafischen Benutzeroberfläche »Intuition«. Aus dem Inhalt: Programm-Module, Variablendeklaration, Strukturanweisungen, Prozeduren, lokale und externe Module, Verwendung von Zeigern, systemnahe Programmierung, Coroutinen (Verarbeitung von parallelen Prozessen), Programmierung unter Intuition (Screens, Windows, Gadgets, Requester).
1988, 350 Seiten, inkl. Disk.
Bestell-Nr. 90554
ISBN 3-89090-554-4
DM 69,–
(sFr 63,50/öS 538,-)

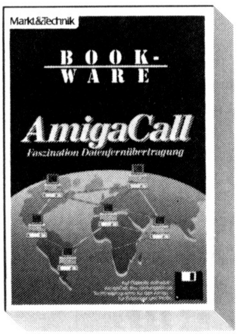
Markt&Technik
Zeitschriften · Bücher
Software · Schulung

Markt&Technik-Produkte erhalten Sie bei Ihrem Buchhändler, in Computer-Fachgeschäften oder in den Fachabteilungen der Warenhäuser.

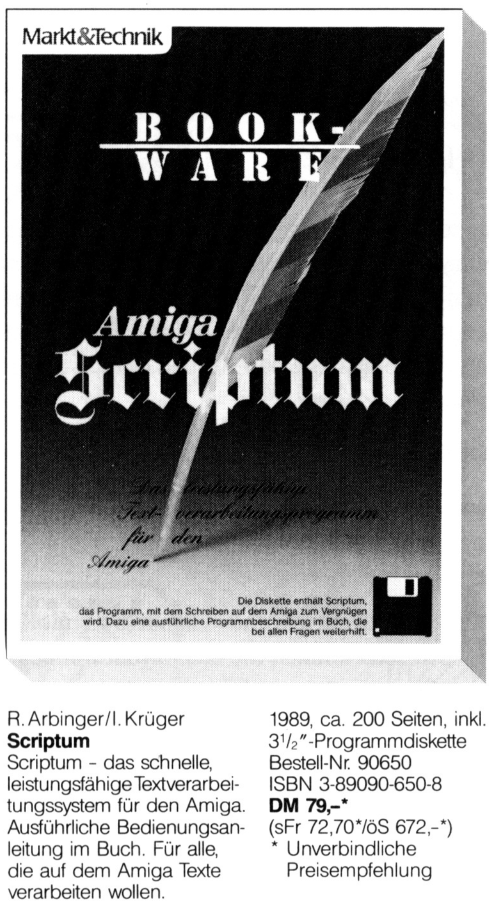
428/901

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

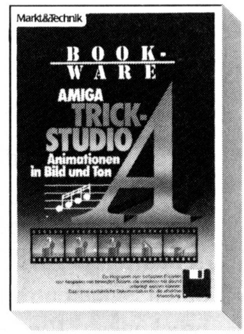
Bücher zum Amiga



Atlantis
AmigaCall
Treten Sie ein in die faszinierende Welt der Datenfernübertragung. Kommunizieren Sie über Mailboxen mit erfahrenen Computer-Anwendern, die Ihnen bei Ihren Problemen weiterhelfen können, oder Sie erhalten auf diesem Wege leistungsfähige Public-Domain-Software. AmigaCall nimmt Ihnen die meiste Arbeit ab. Schließen Sie Ihr Modem oder Ihren Akustikkoppler an, starten Sie AmigaCall - und auf geht's.
1988, 133 Seiten, inkl. $3^{1}/_{2}$″-Programmdiskette
Bestell-Nr. 90716
ISBN 3-89090-716-4
DM 99,-*
(sFr 91,-*/öS 842,-*)



R. Arbinger/I. Krüger
Scriptum
Scriptum - das schnelle, leistungsfähige Textverarbeitungssystem für den Amiga. Ausführliche Bedienungsanleitung im Buch. Für alle, die auf dem Amiga Texte verarbeiten wollen.
1989, ca. 200 Seiten, inkl. $3^{1}/_{2}$″-Programmdiskette
Bestell-Nr. 90650
ISBN 3-89090-650-8
DM 79,-*
(sFr 72,70*/öS 672,-*)
* Unverbindliche Preisempfehlung



Atlantis
Trickstudio A
Ob Sie Computerfilm-Pionier sind oder Trickprofi, ob Sie von Walt Disney inspiriert sind oder einfach nur einen guten Lehrfilm für technische Abläufe benötigen: Mit Trickstudio A können Sie Ihre eigenen Trickfilme erstellen und diese mit Sound und Geräuschen untermalen.
Wie wäre es also mit einem Stummfilm-Slapstick, einem Krimi oder einem Werbefilm für Ihr Schaufenster? Dazu Ihre Lieblingsmusik oder digitalisierte Stimmen? Entwerfen Sie die Einzelbilder, z. B. mit Deluxe Paint, erstellen Sie eine Sounddatei und dann: Klappe - Film; die erste.
1988, 86 Seiten, inkl. $3^{1}/_{2}$″-Programmdiskette
Bestell-Nr. 90715
ISBN 3-89090-715-6
DM 99,-*
(sFr 91,-*/öS 842,-*)

Markt&Technik
Zeitschriften · Bücher
Software · Schulung

427/901

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Anleitung zur Verwendung der Platinenlayouts

Bitte prüfen Sie die Originallayouts!

Wegen technisch bedingter Abweichungen beim Druck des Buches ist es nicht immer gewährleistet, daß die Layouts hundertprozentig in Ordnung sind. Sie erkennen Fehler durch Sichtkontrolle der einzelnen Bahnen nach offensichtlich unerwünschten Überbrückungen. Eine gewollte Verbindung zwischen zwei Bahnen muß klar und deutlich sichtbar sein. Zur Kontrolle können Sie den Aufbau des Layouts mit der Baubeschreibung im Text vergleichen.

Die Layouts können auf mehrere Arten verarbeitet werden.

1. Methode

Schneiden Sie das gewünschte Layout aus und behandeln es mit Klarpaus-Spray, so daß das Papier transparent wird. Jetzt können Sie das Layout mit der bedruckten Seite auf die Platine legen und das Ganze belichten. Weitere Arbeitsanweisungen finden Sie im Anhang A.
Vorsicht, bei dieser Methode kann das Layout nur einmal verwendet werden!

2. Methode

Kopieren Sie das Layout auf Folie. Entsprechendes Material gibt's im Schreibwarenhandel unter dem Stichwort »Overhead-Folie«. Diese Vorlage läßt sich dann beliebig oft verwenden.
Aber auch hier Vorsicht. Beim Kopieren können die Leiterbahnen an Schwärze verlieren und müssen nachgebessert werden.
Sie können das Layout auch auf Papier kopieren und nach Methode 1 verfahren.

Platinenlayouts:

Spiegelverkehrt abgedruckt (Anleitung siehe gegenüber)

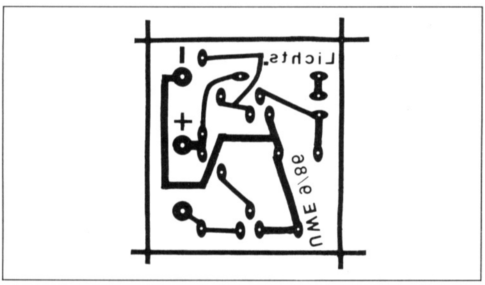


Bild 2.26

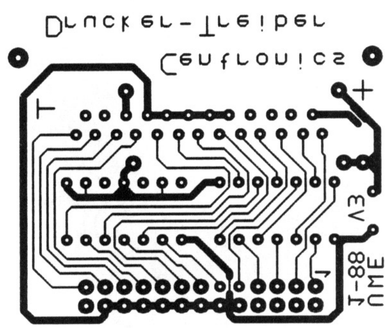


Bild 2.42

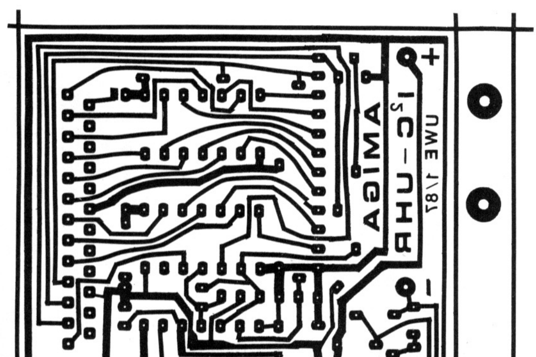


Bild 2.52

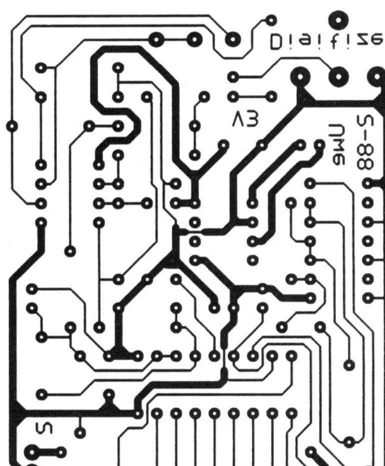


Bild 2.57

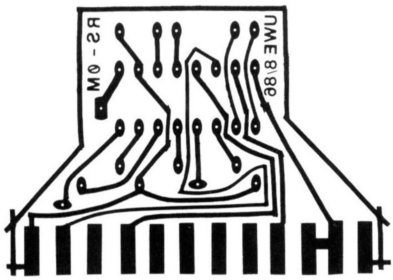


Bild 3.12

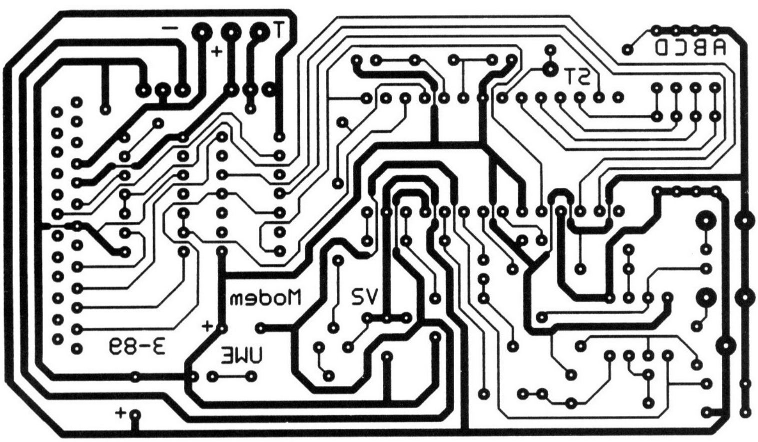


Bild 3.17

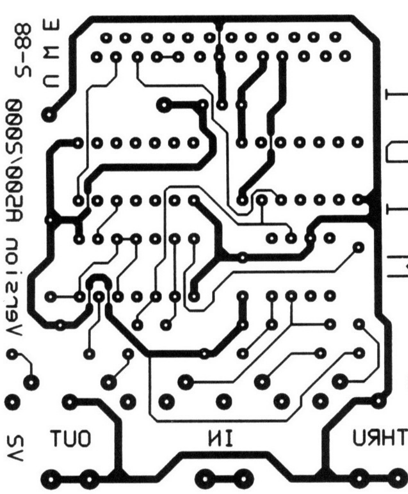


Bild 3.21

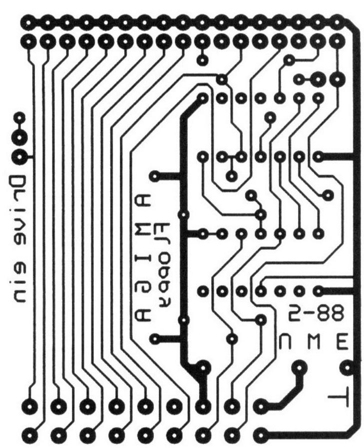
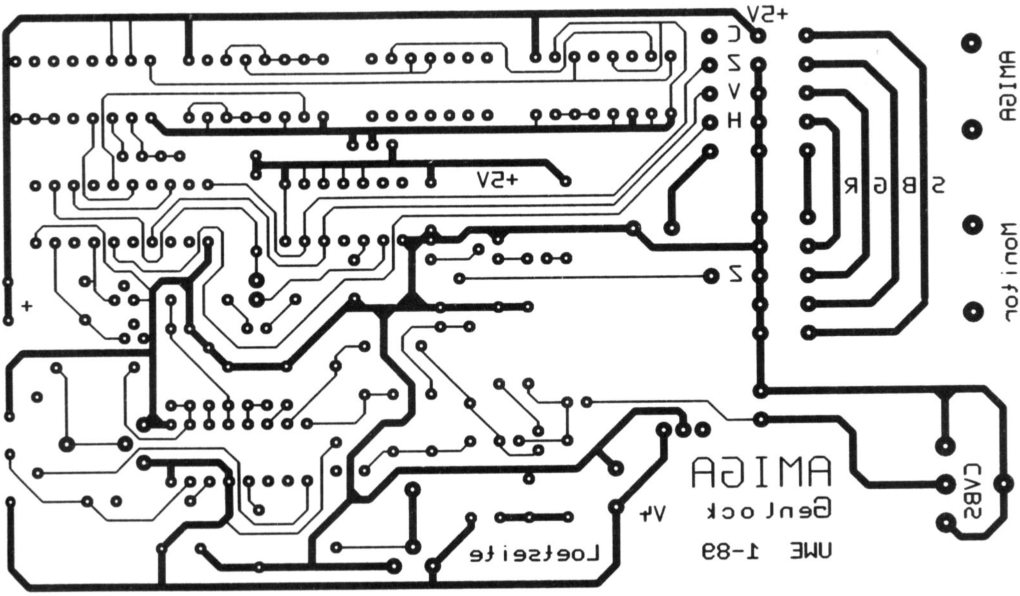


Bild 3.25



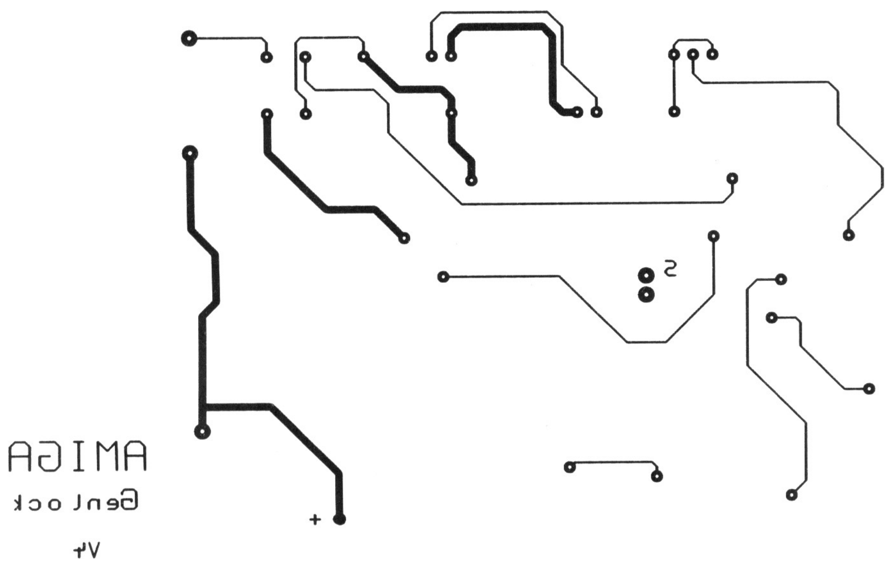


Bild 4.10 Vorder- und Rückseite

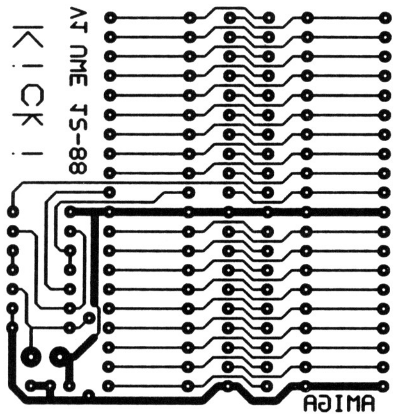


Bild 5.7

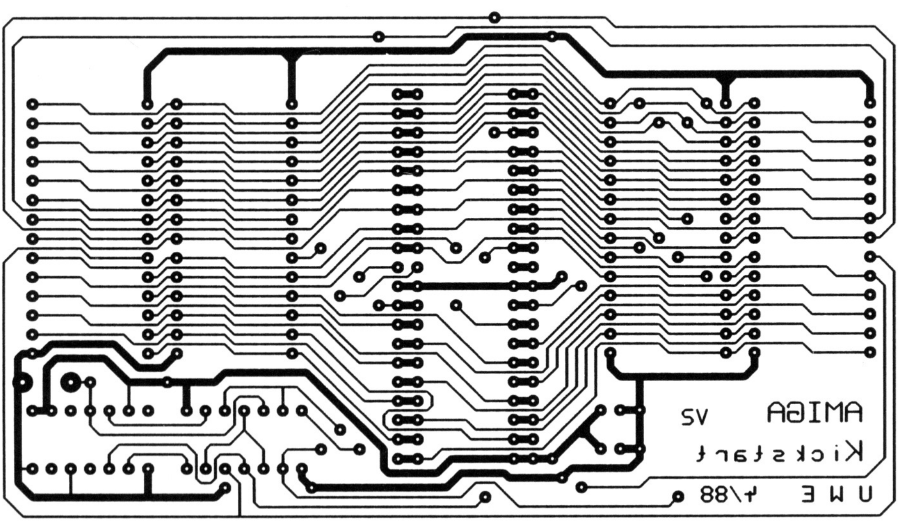


Bild 5.10

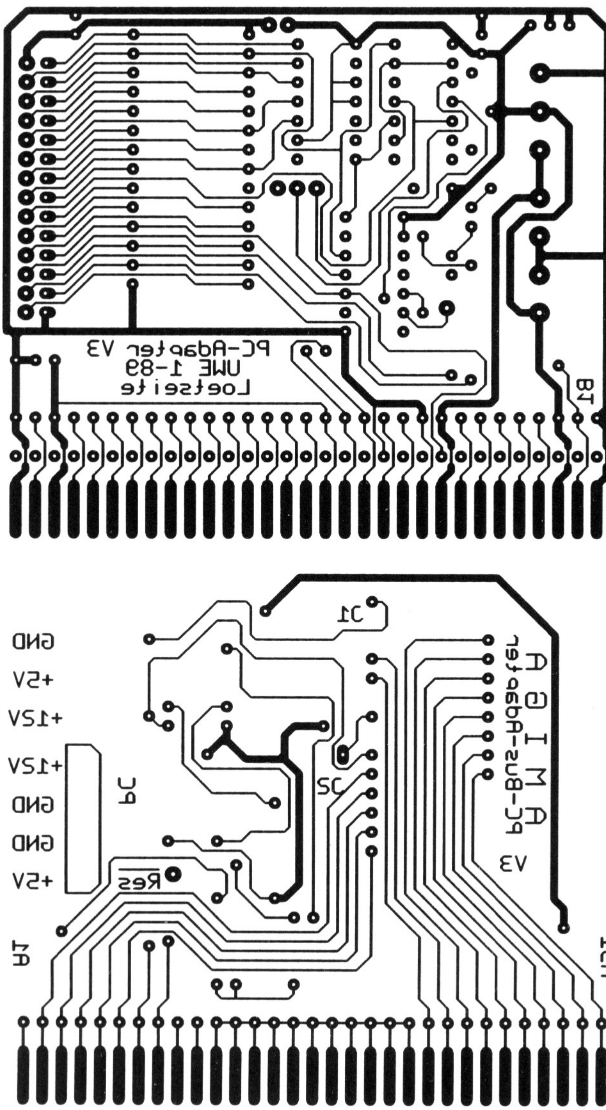


Bild 5.17 Vorder- und Rückseite

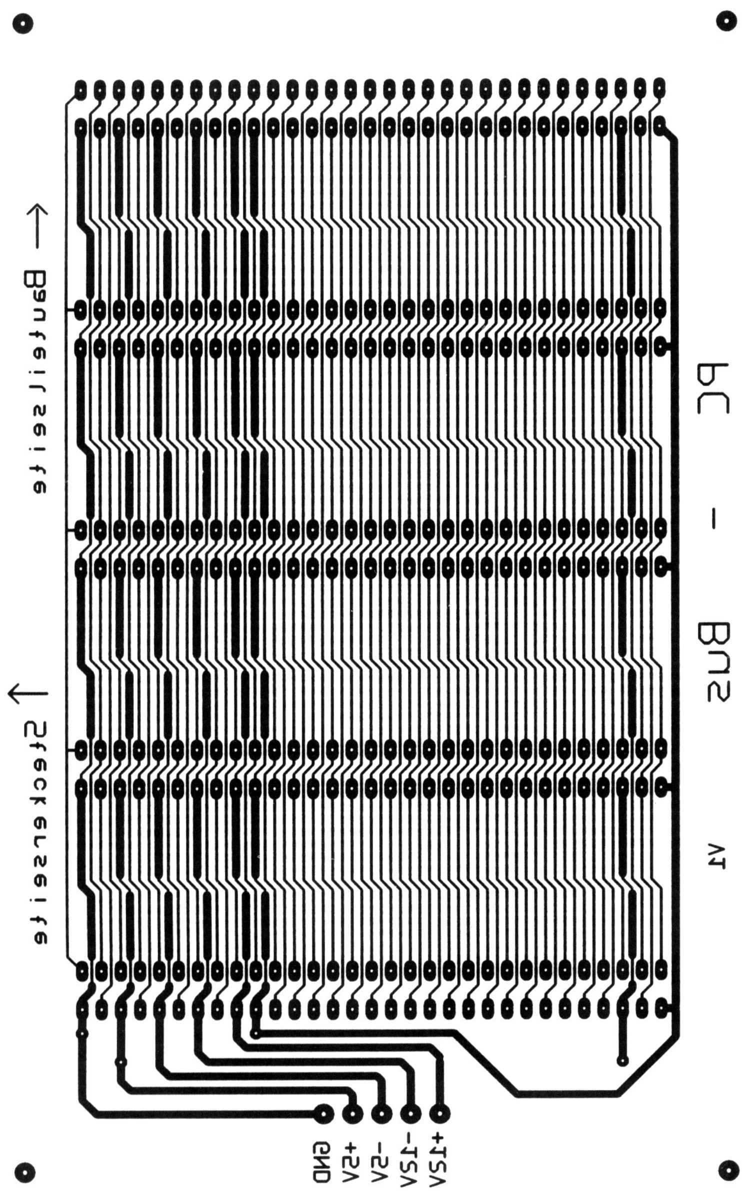


Bild 5.18

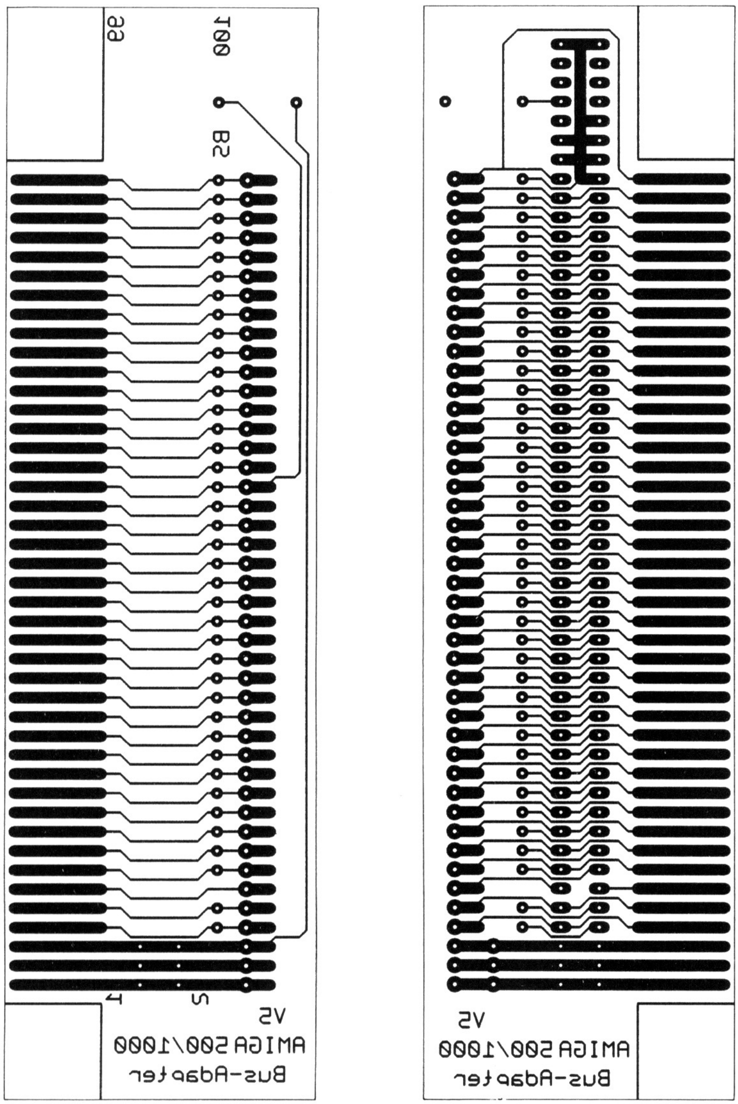
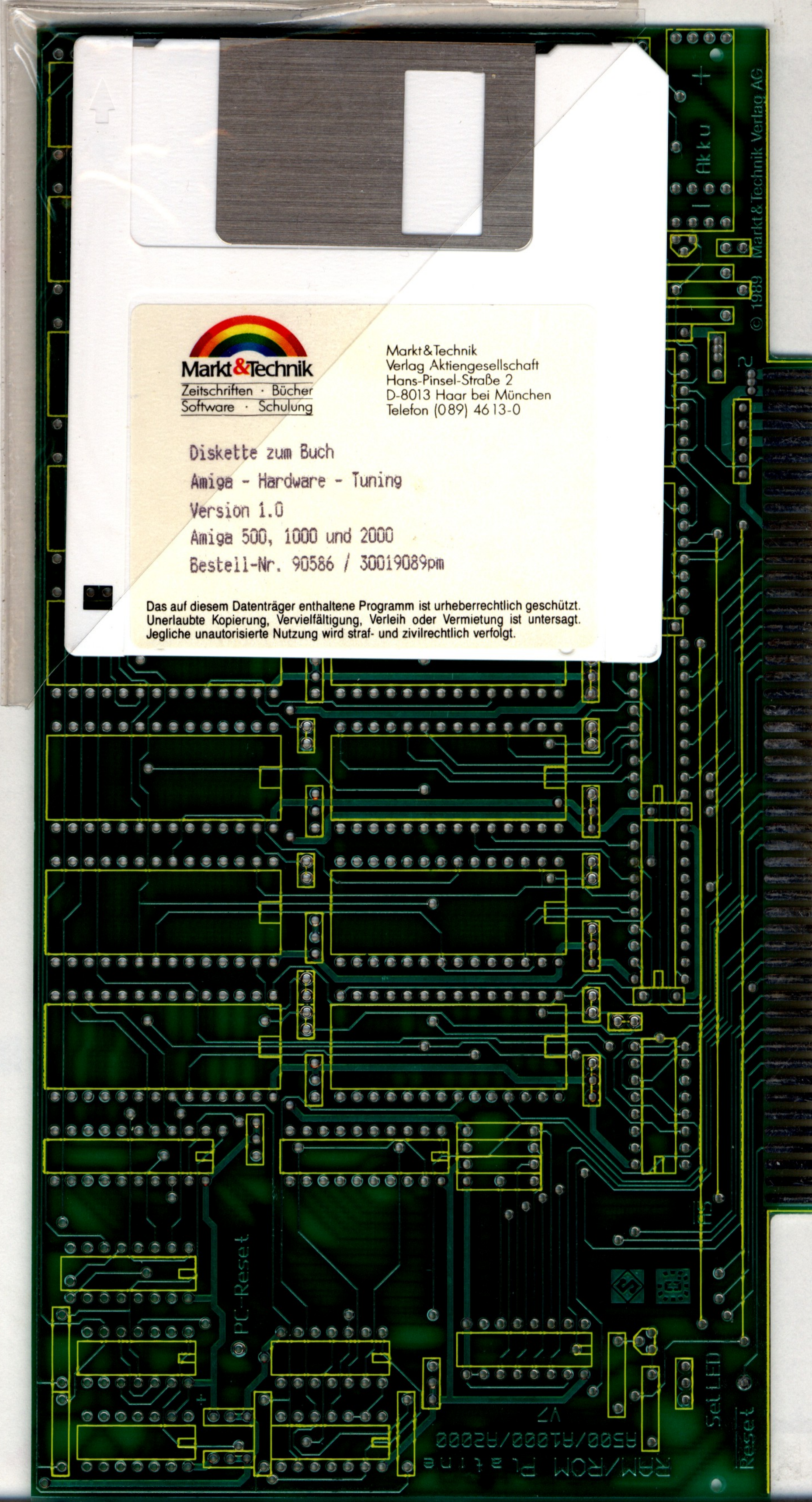


Bild 5.21 Vorder- und Rückseite


Markt&Technik
Zeitschriften · Bücher
Software · Schulung
Markt&Technik
Verlag Aktiengesellschaft
Hans-Pinsel-Straße 2
D-8013 Haar bei München
Telefon (089) 46 13-0
Diskette zum Buch
Amiga - Hardware - Tuning
Version 1.0
Amiga 500, 1000 und 2000
Bestell-Nr. 90586 / 30019089pm
Das auf diesem Datenträger enthaltene Programm ist urheberrechtlich geschützt.
Unerlaubte Kopierung, Vervielfältigung, Verleih oder Vermietung ist untersagt.
Jegliche unautorisierte Nutzung wird straf- und zivilrechtlich verfolgt.
© 1989 Markt&Technik Verlag AG
Akku
PC-Reset
Set LED
Reset

Uwe Gerlach
Christian Hochberger

Amiga-Hardware-Tuning

Die Autoren:

UWE GERLACH und CHRISTIAN HOCHBERGER sind Kenner der Computertechnik, besonders der Amiga-Technik. Sie haben neben ihrem Studium viel Entwicklungsarbeit für mehrere Firmen geleistet und ihr Wissen nun in dieses Buch eingebracht.

Die Amiga 500/1000/2000 von Commodore haben sich zu den beliebtesten Heimcomputern entwickelt. Ausschlaggebend dafür waren die unübertroffenen Grafikfähigkeiten, die der Computer in seiner Preisklasse bietet, die große Anzahl an Anwendersoftware und besonders: die vielen witzigen Spiele. Dennoch ist dies einigen Anwendern zu wenig, Sie wollen mehr - mehr Leistung, mehr Möglichkeiten, mehr Schnittstellen.
Hier setzt dieses Buch an. Es bringt Sie in die Lage, interessante Zusatzgeräte für den Amiga selbst zu bauen. Druckertreiber, Digitizer und Echtzeituhr sind die kleineren Schaltungen, die vorgestellt werden. Dabei gehen die Autoren zuerst auf die Theorie ein, erklären die Wirkungsweise und Schaltungstechnik und kommen dann zur Bauanleitung mit Layout, Bestückungsplan und Stückliste. Für den praktischen Einsatz sind die entsprechenden Steuerprogramme auf Diskette enthalten. So finden Sie alle Informationen, die für einen Eigenbau notwendig sind, in diesem Buch.
Die große, im Buch enthaltene, Platine ist Basis für eine akkugepufferte RAM-/ROM-Karte mit 1 Mbyte Speicherplatz. Sie ist bootfähig und resetfest und läßt superschnelle Zugriffe zum Beispiel auf CLI-Kommandos zu.
Weitere hochkarätige Bausätze sind ein Genlock-Interface, um Bilder vom Videorecorder oder vom Fernseher zu überspielen; ein EPROMer, um spezielle Daten oder Software schnell verfügbar zu haben, oder eine Kickstart-Umschaltplatine.
Folgende Selbstbauprojekte sind im Buch enthalten:

- IC-Bus mit Echtzeituhr
- Drucker-Schnittstelle
- Akustikkoppler
- Digitizer
- Schallwandler
- MIDI-Interface
- Disk-Adapter
- PC-Bus-Adapter
- Modem
- Anschluß von Fremdlaufwerken
- EPROM-Programmierer
- RAM-/ROM-Karte
- Kickstart-Umschaltplatine
- Genlock-Interface

Die Platinen-Layouts (erstellt mit NEWIO von Alphatron) sind auf einem gesonderten Bogen dem Buch beigelegt.
Mit den Informationen in diesem Buch können Sie auf billige Art und Weise zum Amiga-Tuner werden und langgehegte Wünsche an Hard- und Software selbst erfüllen.

Hardware-Anforderungen:
Amiga 500/1000/2000

ISB N 3-89090-586-2

Markt&Technik

4 001057 905863

DM 98,-
sFr 90,20
öS 764,-